VersaMax™ Workshop
Student Guide

GFN-034 *April 2000*

This document is based on information available at the time of its publication. While efforts have been made to be accurate, the information contained herein does not purport to cover all details or variations in hardware or software, nor to provide for every possible contingency in connection with installation, operation, or maintenance. Features may be described herein which are not present in all hardware and software systems. GE Fanuc Automation assumes no obligation of notice to holders of this document with respect to changes subsequently made.

GE Fanuc Automation makes no representation or warranty, expressed, implied, or statutory with respect to, and assumes no responsibility for the accuracy, completeness, sufficiency, or usefulness of the information contained herein. No warranties of merchantability or fitness for purpose shall apply.

The following are trademarks of GE Fanuc Automation North America, Inc.

Alarm Master	Genius	PROMACRO	Series Six
CIMPLICITY	Helpmate	PowerMotion	Series Three
CIMPLICITY 90–ADS	Logicmaster	PowerTRAC	VersaMax
CIMSTAR	Modelmaster	Series 90	VersaPro
Field Control	Motion Mate	Series Five	VuMaster
GEnet	ProLoop	Series One	Workmaster

©Copyright 2000 GE Fanuc Automation North America, Inc.
All Rights Reserved.

Contents in Brief

Unit 1	**System Overview:** Introduces PLCs, program solving, key definitions, memory reference definition, VersaMax PLC, programming devices, and instruction sets.
Unit 2	**Configuration Lab:** Step-by-step instructions for configuring VersaMax hardware.
Unit 3	**Programming Lab:** Step-by-step instructions for programming the VersaMax PLC.
Appendix A	**VersaMax Menus and Toolbars**

Related Publications

GFK-1503: VersaMax PLC User's Manual

GFK-1504: VersaMax Modules, Power Supplies, & Carriers Manual

GFK-1535: VersaMax Genius® Network Interface Unit User's Manual

GFK-1534: VersaMax Profibus-DP Network Interface Unit User's Manual

GFK-1533: VersaMax DeviceNet™ Network Interface Unit User's Manual

GEK-90486-1: Genius I/O System and Communications Manual

Contents

UNIT ONE - SYSTEM OVERVIEW 5

INTRODUCTION TO PLCs 5
PROGRAM SOLVING 6
KEY DEFINITIONS 7
MEMORY REFERENCE DEFINITION 9
VERSAMAX™ PLC 10
PROGRAMMING DEVICES 12
INSTRUCTION SET 14

UNIT TWO – CONFIGURATION LAB 25

USING VERSAPRO 26
THE WORKBENCH WINDOW 27
SETTING WORKBENCH OPTIONS 27
SETTING DISPLAY OPTIONS 29
CREATING A NEW FOLDER 30
TO CREATE A NEW FOLDER 30
CUSTOMIZING THE WORKSPACE 33
VERSAMAX CONFIGURATION 35
STORING THE CONFIGURATION TO THE VERSAMAX 41

UNIT THREE - PROGRAMMING LAB 45

GETTING STARTED 46
ENTERING RELAY LADDER LOGIC 49
STORING THE CONFIGURATION TO THE VERSAMAX PLC 58
MONITORING THE LOGIC PROGRAM 60
OVERRIDE AND TOGGLE 60
REFERENCE VIEW TABLES 61
VARIABLE VIEW TABLES 63
FAULT TABLES 64
CREATING, INSERTING BLOCKS 66
CREATING A NEW BLOCK 66
ENTERING INSTRUCTION LIST LOGIC 67
INSERTING AND EDITING OBJECTS 69
PRINTING THE PROGRAM LOGIC 70
PRINTING THE CONFIGURATION 71

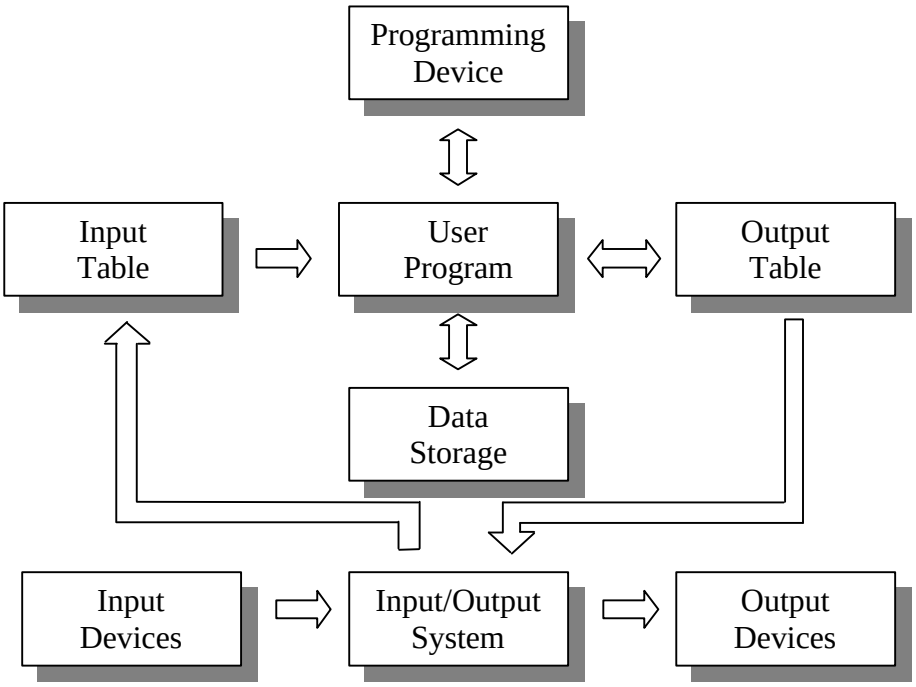
APPENDIX A 1

VERSAPRO MENU BARS 2
FILE MENU 2
EDIT MENU 3
VIEW MENU 4
INSERT MENU 5
FOLDER MENU 5
PLC MENU 6
TOOLS MENU 6
WINDOW MENU 7
HELP MENU 7
TOOLBARS 7
LADDER TOOLBAR 8
VIEW TOOLBAR 9
FOLDER TOOLBAR 10
PLC TOOLBAR 10
FUNCTION TOOLBAR 11

Unit ONE - System Overview

INTRODUCTION TO PLCs

A Programmable Logic Controller (PLC) is an industrial computer that accepts inputs from switches and sensors, evaluates these in accordance with a stored program, and generates outputs to control machines and processes.



PROGRAM SOLVING

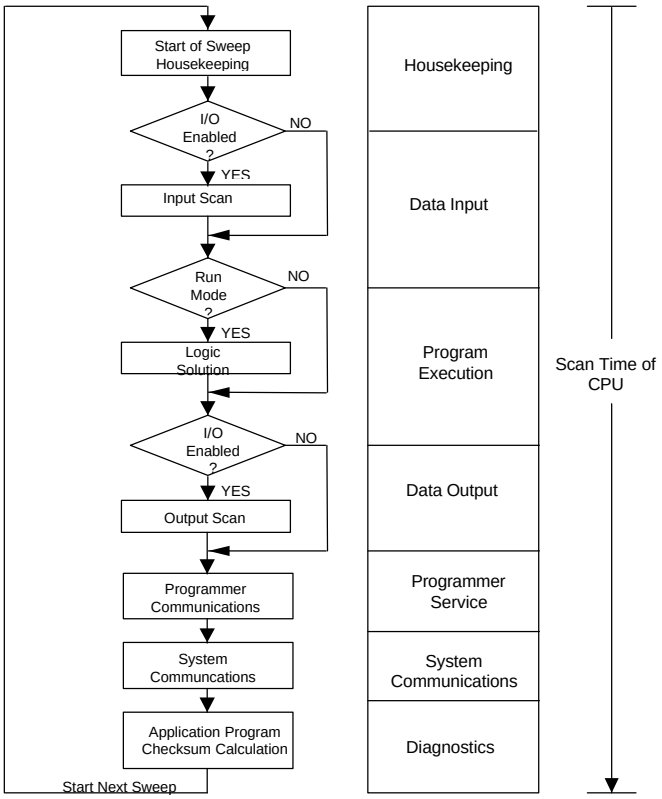
The application program in a PLC executes repeatedly. In addition to executing the application program, the PLC CPU regularly obtains data from input devices, sends data to output devices, performs internal housekeeping, and performs communications tasks. This sequence of operations is called the **sweep**.

- The basic operating mode of the PLC is called **Standard Sweep** mode. In this mode, the CPU performs all parts of its sweep normally. Each sweep executes as quickly as possible with a different amount of time consumed each sweep.

- The PLC may instead operate in **Constant Sweep Time** mode. In this mode, the CPU performs the same series of actions but each sweep takes the same amount of time.

The PLC may also be in either of two Stop modes:

- Stop with I/O Disabled mode
- Stop with I/O Enabled mode



KEY DEFINITIONS

- *Memory*. The part of a programmable controller where data and instructions are stored either temporarily or semipermanently. The control program is stored in memory.

- *Inputs.* Any input device connected equipment that will supply information to the central processing unit, such as, switches, buttons, limit switches, sensors, etc..

- *Outputs.* Any output device connected to equipment that will receive information or instructions from the central processing unit, such as control devices like motors, solenoids, lights, alarms, etc.

- *I/O.* Abbreviation for Inputs and Outputs

- *Address.* An alphanumeric value that uniquely identifies where data is stored. (Ex. %I0001 is input 1, %Q0003 is output 3, and %R0100 is Register 100)

- *High Speed Counter.* Allows rapid pulses that are faster than the scan time of the PLC to be counted independently.

- *Encoder.* A rotary device which transmits position information. The pulses are feed into a high speed counter on the PLC

- *Status Table.* The status table is part of the CPU that stores the status of all the Inputs (%I000x), Outputs (%Q000x), Registers (%R000x) etc.

- *Normally Open Contact*. A ladder logic symbol that will allow logic continuity (flow) if the reference input is a logic "1" when evaluated.

- *Normally Closed Contact.* A ladder logic symbol that will allow logic continuity (flow) if the reference input is a logic "0" when evaluated.

- *PID Proportional-Integral Derivative.* A mathematical formula that provides a closed loop control of a process. Inputs and outputs are continuously variable and typically will be analog signals.

- *Scan Time*. The time required by the processor to read all inputs, execute the control program, and update I/O, evaluate and execute the control logic. The program scan repeats continuously while the processor is in the run mode.

- *Register*. A temporary storage device for various types of information and data (e.g. Timer/Counter values) In the PLC a register is normally 16 bits wide or hold a number up to 32,768.

- *Flash Memory*. Provides a non-volatile user-program storage and system firmware. Does not require battery back-up.

- *Din Rail*. 35mm DIN rail can be used to mount the VersaMax PLC

- *PWM*. PWM stands for Pulse Width Modulation used in motion control where position is critical

- *SNP*. SNP (Series Ninety Protocol) is the protocol required to talk to the GE Fanuc Series 90 Family of PLCs and VersaMax

- *P/S*. Power supply

- *Bit*. One binary digit. The smallest unit of binary information. A bit can have a value of "1" or "0"

- *Byte*. A group of adjacent bits usually operated upon as a unit, such as when moving data to and from memory. There are 8 bits in a byte.

- *Word*. The unit of binary digits (bits) operated on at a time by the central processing unit when it is performing an instruction or operating on data. A word is usually composed of a fixed number of bits. One word is equal to (2) bytes or 16 bits.

- *AND*. A Boolean operation that yields a logic "1" output if all inputs are "1" and a logic "0" if any input is a "0"

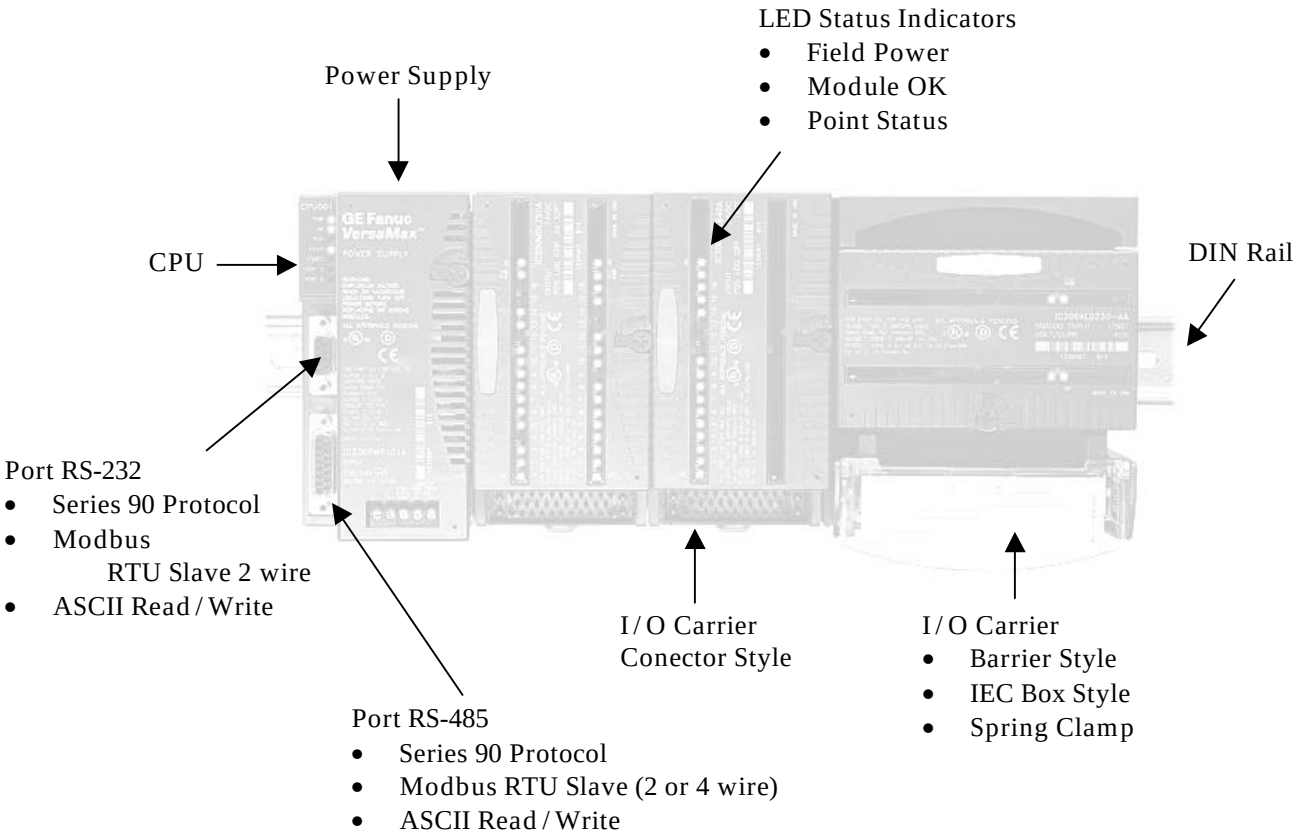
- *OR*. A Boolean operation that yields a logic "1" output if any inputs are a "1" and a "0" if all inputs are a "0"

MEMORY REFERENCE DEFINITION

Type	Definition	Function
%I	Discrete Reference input Point (bit)	The state of the input as detected during the last input scan.
%Q	Discrete Reference Output point (bit)	The state of the output as last set by application program.
%M	User Internal (bit)	Internal coil used for boolean logic when the result of a rung is only required to be used later in the program as conditional logic.
%T	Temporary (bit)	Internal coil - similar to %M reference except that it is non-retentive.
%S	System Discretes (S, SA, SB, SC) (bit)	Includes system bits used internally by the CPU, fault bits for holding system fault data and reserved bits for future system expansion.
%R	Register Reference (Word)	%R is used to assign system register references, which will store program data such as the result of calculations and timer/counter data.
%AI	Analog Input Register (Word)	The analog input register holds the value of one analog input or other value
%AQ	Analog Output Register (Word)	An analog output register holds the value of one analog output or other value
%G	Global Data References	These references are used to access data shared among several PLCs

VERSAMAX™ PLC

The VersaMax PLC features a compact, rackless design and DIN-rail mounting.



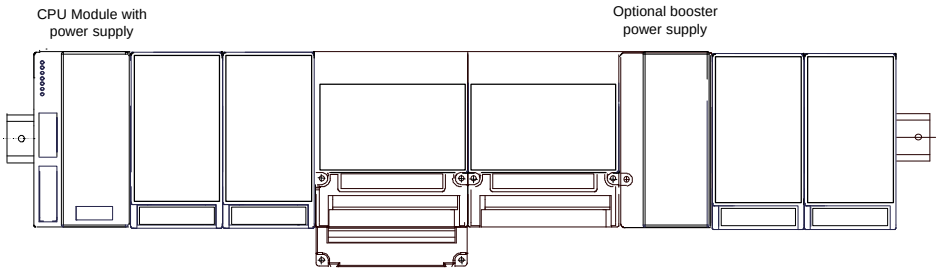
The CPU, with its powerful programming Instruction Set and advanced features serves up to eight I/O and option modules per "rack" and up to 8 racks in the PLC system. Expansion racks can be located up to 750 meters from the main VersaMax PLC rack. Expansion racks can include any VersaMax I/O, option, or communications module.

Power for module operation is provided by a power supply that mounts directly on the CPU itself. Additional power supplies can be included in the system if needed for modules with high current requirements.

I/O modules mount on individual "carriers". Carriers install on the DIN rail and provide backplane communications and field wiring terminals for the module. A variety of carrier styles provide mounting and field wiring flexibility.

The illustration below shows a local system with six I/O modules. The modules can be mounted on any combination of carrier styles, as shown. On connector-type carriers, modules are oriented vertically with respect to the DIN rail.

Connector-type carriers have a 32-pin connector for attaching an I/O cable. Actual field wiring terminals for these carriers are located on special interposing terminal units (not shown). On terminal-type carriers, modules are mounted horizontally with respect to the DIN rail. Terminal-type carriers provide 32 screw-down terminals for direct connection of field wiring. Additional auxiliary I/O terminals can be added if more terminals are needed. One such auxiliary terminal unit is shown below.



Designed for industrial and commercial automation, VersaMax meets UL, CUL, CE, Class1 Zone 2 and Class I Division 2 requirements.

Autoconfiguration at powerup provides automatic addressing for the modules in the system. Modules have a default set of features that are suitable for a wide range of applications.

CPU FEATURES

The VersaMax CPU provides a broad Instruction Set, with programming in Ladder Diagram and Sequential Function Chart format. Programs can include floating point (real) data functions.

CPU models IC200CPU001 and 002 have two serial ports for embedded RS232 and RS485 communications. Both ports are software-configurable for SNP Slave or RTU Slave operation. 4-wire and 2-wire RTU are supported. In addition, the Serial I/O feature allows an application program to control the read/write activities of a port to implement most serial protocols.

PROGRAM MEMORY

Application programs are created using the programming software and transferred to the PLC. Programs are stored in the CPU's battery-backed memory. The amount of memory available for application program storage depends on the CPU model.

CPU Model	Program Memory
IC200CPU001	12kB

CPU Model	Program Memory
IC200CPU002	20kB

SUBROUTINES

The program can consist of one main program that executes completely during each CPU sweep, or the program can be divided into subroutines.

Subroutines can simplify programming and reduce the overall amount of logic. Each subroutine can be called as needed.

Subroutine blocks can be locked and unlocked from the programming software for varying levels of program security. There are four levels of locking.

SYSTEM STATUS REFERENCES

The VersaMax CPU provides a full set of system status references. The CPU automatically updates these pre-defined, named memory locations with status information. They can be accessed as needed by the program logic to check for and respond to changes in system conditions. These special status references include time-tick references named T_10MS, T_100MS, T_SEC, and T_MIN and convenience references such as FST_SCN (first scan), ALW_ON (always on), and ALW_OFF (always off). See chapter 7 for more information about system status references.

PROGRAMMING DEVICES

VersaPro Programming Software is the programmer for the full range of Series 90™-30 and VersaMax CPUs. VersaPro's small computer footprint and intuitive programming interface make it easy to program and connect to your PLC. VersaPro provides both Relay Ladder Diagram (RLD) and Instruction List (IL) 0logic editors. VersaPro imports Logicmaster™ 90 and Control programs, making it easy for existing customers to convert their programs to the VersaPro software platform. In addition, VersaPro offers numerous capabilities for interfacing to external Windows applications.

VersaPro software was designed to run well on a low-end Windows 95/98/NT PC.

The application is composed of the following components that are used during the creation of PLC programs:

- *Browser:* this simple navigation tool allows the user to navigate quickly through the application. In addition, this view is an ideal location to add,

delete or select components from the folder for activation, printing, or simply to display their properties.

- *Variable Declaration Table (VDT):* The VDT is the location where all variable data resides. Variables can be created in many places, but all of that information is displayed and managed in this single view.

- *Information Window:* This window provides a detailed description of the syntax checks, import and export operations and all find/search and replace actions. By double-clicking on entries in this window, the applications navigates to the appropriate place to allow editing of that item whether it is an element in a rung of logic or a variable name.

- *Logic Editor:* The RLD and IL Editors are the heart of the application. The programmer can easily enter graphical ladder logic using either the mouse or keyboard. Language editor windows are free-form editors allowing the user to quickly enter logic on the screen without interruption. The program can be checked for validity at any point simply by clicking on the Check Block icon on the toolbar to make the process as efficient as possible.

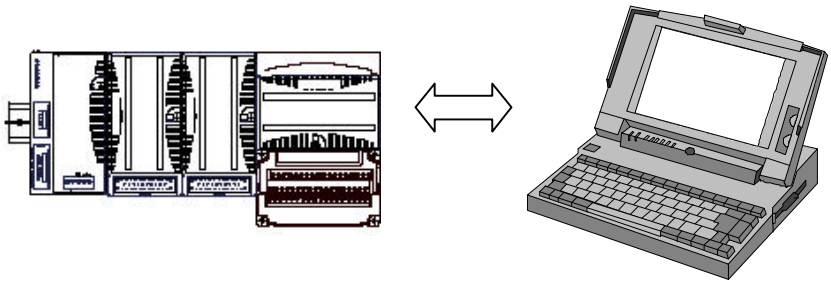
- *Hardware Configuration Window*: VersaPro's graphical hardware configuration utility makes hardware configuration a snap, this utility boosts productivity by constantly checking the configuration for errors, and warning the user.

- *Monitoring Table:* VersaPro provides two types of data monitoring tables, Reference View Tables and Variable View Tables. The format of both these tables can be saved in the Browser for easy retrieval.

    - ✓ *Variable View Tables (VVT)*: These configurable tables allow the user to see the properties of user-defined groups of variables. Variables can simply be dragged from the logic editor or the VDT and dropped in the VVT. Multiple VVTs can be saved in the Browser.

    - ✓ *Reference View Table (RVT)*: The Reference View Table allows the user to view sections of PLC memory in a compact table format to maximize the amount of data available for troubleshooting. Like the VVT, the RVT is configurable and can be saved in the Browser.

- *Fault Table*: The fault table runs as a separate window, and provides detailed diagnostic view of the PLC system. In addition, the table gives the user the ability to clear faults in the PLC.



INSTRUCTION SET

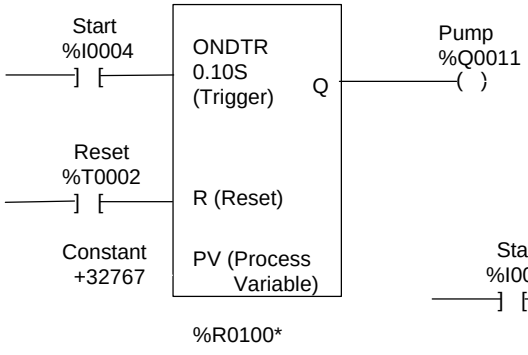
The VersaMax PLC CPU provides a powerful Instruction Set for building application programs. As a guide to the programming capabilities of the VersaMax PLC, all of the relays, coils, functions, and other elements of the Instruction Set are summarized on the following pages. Complete reference information is included in the documentation and online help for the programming software.

Contacts Normally Open, Normally Closed	***Coils*** Normally Open Negated Positive & Negative Transition SET / RESET Retentive SET / Retentive RESET Negated Retentive / Retentive
Timers and Counters On-Delay Stopwatch Timer Off-Delay Timer On-Delay Timer Up Counter Down Counter	***Math Functions*** Addition / Subtraction / Multiplication / Division Modulo Division Trigonometric: Sine, Cosine, Tangent, Inverse Sine, Inverse Cosine, Inverse Tangent Convert to Degrees / Convert to Radians Square Root Base 10 Logarithm /Natural Logarithm Power of e
Relational Functions Equal Not Equal Greater Than Greater Than or Equal To Less Than Less Than or Equal To	***Bit Operation Functions*** Logical AND / OR / Exclusive OR / Invert Shift Left / Shift Right Rotate Left / Rotate Right Bit Test / Bit Set / Bit Clear / Bit Position Masked Compare

Range	
Data Move Functions Move Block Move Block Clear Shift Register Bit Sequencer Communication Request	***Table Functions*** Array Move Search: Equal / Not Equal / Greater Than / Greater Than or Equal /Less Than /Less Than or Equal
Conversion Functions Convert Integer to BCD-4 Convert Real to Word Convert BCD-4 or Real to Integer Convert BCD-4 or Real to Double-Precision Integer Convert Integer, Double-Precision Integer, BCD-4, or Word to Real Truncate Real to Integer Truncate Real to Double-Precision Integer	***Control Functions*** Call Do I/O Independent PID Algorithm ISA PID Algorithm Temporary End of Logic Comment Service Request Nested Master Control Relay Jump

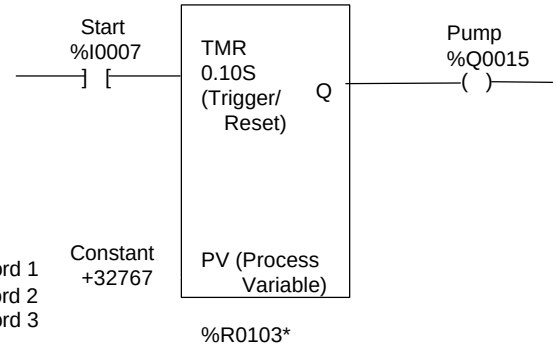
Timers

Timers: On-Delay



When the accumulated value is greater than or equal to the preset variable "Q" will be energized.

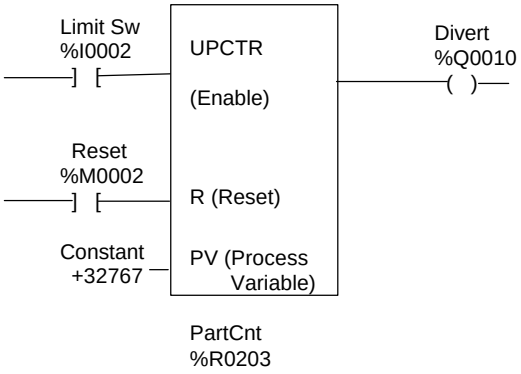
Timer



*Note: Timers and Counters Consecutive
Current Value = Word 1
Preset Value = Word 2
Control Word = Word 3
Caution - Never Overlap

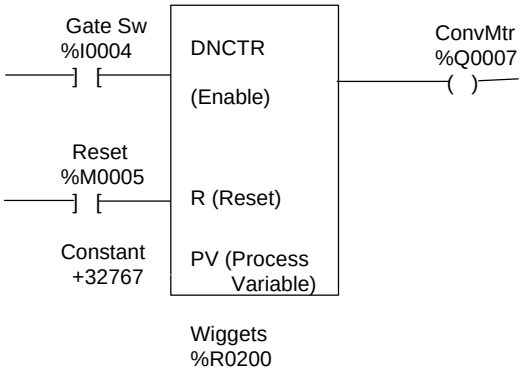
Counters

Counters: Up Counter



Note: When the Up counter reset is ON, current value of the counter is reset Each time the enable input transitions from OFF to ON, the current value is incremented by 1 (Reset line must be OFF to count). The current value can be incremented past the preset value PV. The output is ON whenever the current value is greater than or equal to the preset value.

Counters: Down Counter



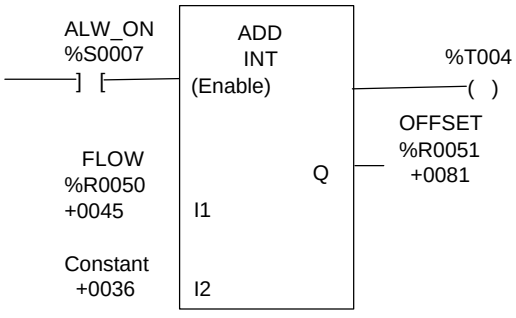
Note: When the Down Counter reset is ON, the current value of the counter is reset to PV. Each time the enable input transitions from OFF to ON, the current value is decremented by 1 (Reset line must be OFF to count). The output is ON whenever the current value is less than or equal to the 0.

Math Functions

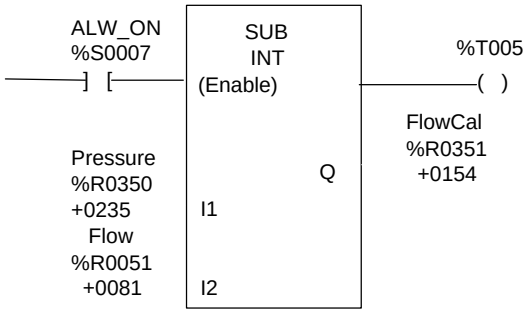
Math - Add, Subtraction, Multiplication, Division (Division rounds down to the closest integer), Modulo Division and Square Root

Data Types - Int (Signed Integer) and DINT (Double precision signed integer)

Math: Add



Math: Subtract



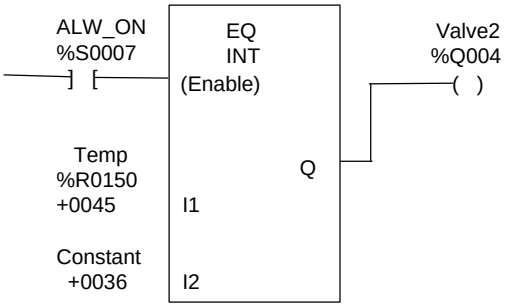
Note: When a function receives power flow, the appropriate math function is performed on input parameters I1 and I2. These parameters must be the same data type. Output Q is the same data type as I1 and I2

Relational

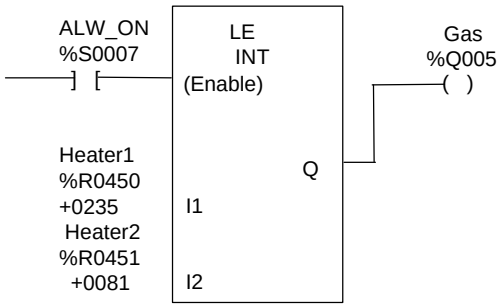
Relational - Equal, Not Equal, Greater Than, Greater/Equal, Less Than
Less/Equal, and Range

Data Types - Int (Signed Integer) and DINT (Double precision signed

Relational: Equal



Relational: Less/Equal



Note: Relational functions are used to determine the relation of two values. When the function receives power flow, it compares input parameter I1 to input parameter I2. These parameters must be the same type. If input parameter I1 and I2 match the specified relation, output Q receives power flow and is set ON, otherwise Q is set OFF.

Bit Functions

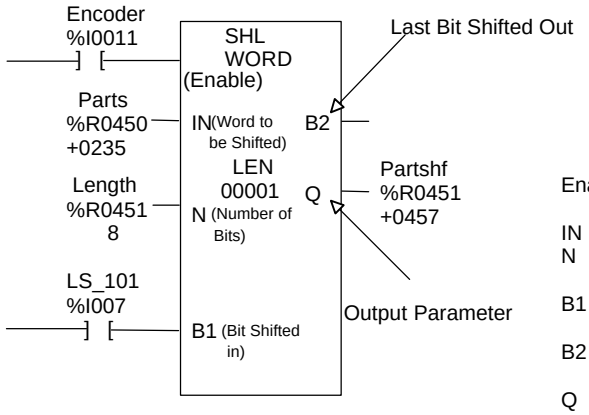
Bit Functions Shift Left/Right, Rotate Left/Right, Logical And, Logical Or, Logical Xor, Logical Inverted Not, Set/Clear
Bit operation Functions perform comparison, logic, and move operations on bit strings. The maximum string length is 256 words or 4096 bits. Bit operation functions require WORD data.

Although data must be specified in 16-bit increment, these functions operate on data as a continuous string of bits, with bit 1 of the first word being the Least Significant Bit(LSB). The last bit of the last word is the Most Significant Bit (MSB).

For example, if you specified three words of data beginning at reference %R0100, it would be operated on as 48 contiguous bits.
Possible applications include material handling and pattern recognition

Shift Left. Used to shift all the bits in a word or group of words to the left/right by a specified number of places. When the shift occurs, the specified number of bits is shifted out of the output string to the left. As bits are shifted out of the high end of the string, the same number of bits is shifted in at the low end.

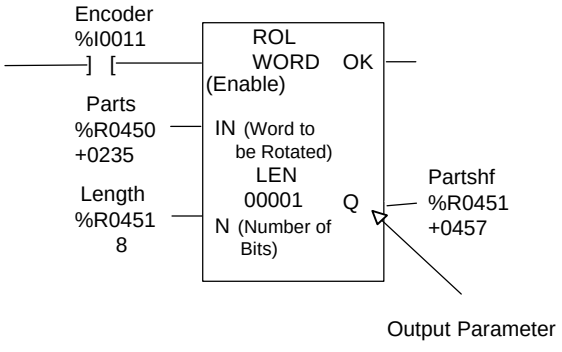
Bit Function: Shift Left



	Description
Enable	When the function is enabled, the shift is performed
IN	IN contains the first word to be shifted.
N	N Contains the number of places (bits) that the array is to be shifted
B1	B1 contains the bit value to be shifted into the array
B2	B2 contains the bit value of the last bit shifted out of the array
Q	Output Q contains the first word of the shifted array
LEN	LEN is the number of words in the array to be shifted

Rotate Left function is used to rotate all the bits in a string a specified number of places to the left. When rotation occurs, the specified number of bits is rotated out of the input string to the left and back into the string on the right. A string length of 1 to 256 words can be selected for the function.

Bit Function: Rotate Left



	Description
Enable	When the function is enabled, the rotation is performed
IN	IN contains the first word to be rotated
N	N Contains the number of places that the array is to be rotated
OK	The ok output is energized when the rotation is energized and the rotation length is not greater than the array size.
Q	Output Q contains the first word of the rotated array
LEN	LEN is the number of words in the array to be rotated

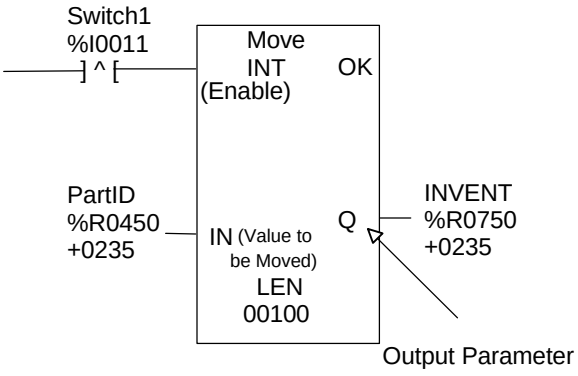
Data Moves

Data Move functions provide basic data move capabilities.

Move	Copy data as individual bits. The maximum length allowed is 256 bits. Data can be moved into a different data type without prior conversion.
Block Move	Copy a block of seven constants to a specified memory location. the constants are input as part of the function.
Block Clear	Replace the content of a block of data with all zeros. The maximum length allowed is 256 words.
Shift Register	Shift one or more data words into a table. The maximum length allowed is 256 words.
Bit Sequencer	Perform a bit sequence shift through an array of bits. The maximum length allowed is 256 words.

MOVE. Use this function to copy data (as individual bits) from one location to another. Because the data is copied in bit format, the new location does not need to be the same data type as the original.

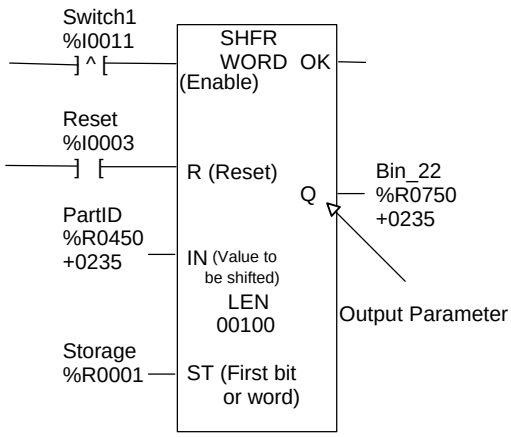
Data Move Function: Move (BIT, INT, WORD)



	Description
Enable	When the function is enabled, the move is performed
IN	IN contains the value to be moved. For MOVE_BIT, any discrete reference may be used; it does not need to be byte aligned. However, 16 bits, beginning with the reference address specified, are displayed on-line.
OK	The ok output is energized when the function is enabled
Q	When the move is performed, the value at IN is written to Q. For MOVE_BIT, any discrete reference may be used; it does not need to be byte aligned. However, 16 bits, beginning with the reference address specified, are displayed on-line.
LEN	LEN specifies the number of words or bits to be moved. For MOVE_WORD and MOVE_INT, LEN must be between 1 and 256 words. For MOVE_BIT, when IN is a constant, LEN must be between 1 and 16 bits.

Shift Register. To shift one or more data words or data bits from a reference location into a specified area of memory. For example, one word might be shifted into an area of memory with a specified length of five words. As a result of this shift, another word of data would be shifted out of the end of the memory area.

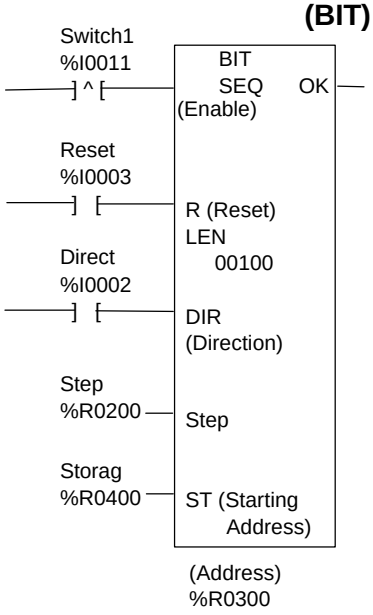
Data Move Function: Shift Register (BIT, WORD)



	Description
Enable	When enable is energized and R is not, the shift is performed
IN	IN contains the value to be shifted into the first bit or word of the shift register. For SHFR_BIT, any discrete reference may be used; it does not need to be byte aligned.
ST	ST contains the first bit or word of the shift register. For SHFT_BIT, any discrete reference may be used; it does not need to be byte aligned.
OK	The ok output is energized when the function is enabled.
Q	Output Q contains the bit or word shifted out of the shift register. For SHFR_BIT, any discrete reference may be used; it does not need to be byte aligned.
LEN	LEN determines the length of the shift register. For SHFR_WORD, LEN must be between 1 and 256 words. For SHFR_ WORD , LEN must be between 1 and 256 bits.

Bit Sequencer (BITSEQ). Performs a bit sequence shift through an array of bits. The BITSEQ function has five input parameters and one output parameter.

Data Move Function: Bit Sequencer (BIT)



	Description
Address	Address is the location of the bit sequencer's current step, length, and the last enable and ok status's.
Enable	When the function is enable, if it was not enabled on the previous sweep and if R is not energized, the bit sequence shift is performed.
R	When R is energized, the bit sequencer's step number is set to the value in STEP (default = 1), and the bit sequencer is filled with zeros, except for the current step number bit.
DIR	When DIR is energized, the bit sequencer's step number is incremented prior to the shift. Otherwise, it is decremented.
STEP	When R is energized, the step number is set to this value.
ST	ST contains the first word of the bit sequencer.
OK	The ok output is energized whenever the function is enabled.
LEN	LEN must be between 1 and 256 bits.

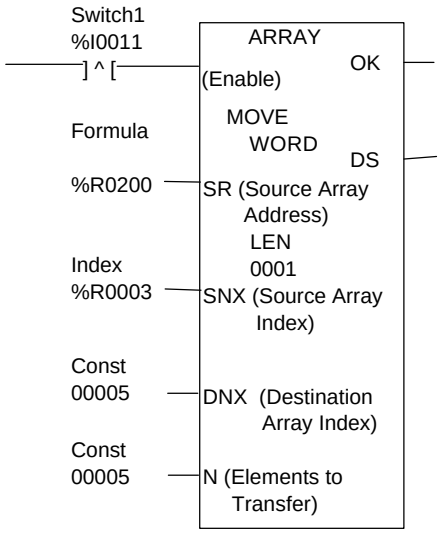
Table Functions

Table functions are used to perform the following functions:

Array Move	Copy a specified number of data elements from a source array to a destination array.
Search Equal	Search for all array values equal to a specified value.
Search Not Equal	Search for all array values not equal to a specified value.
Search Greater Than	Search for all array values greater than to a specified value.
Search Greater Than or Equal	Search for all array values greater than or equal to a specified value.
Search Less Than	Search for all array values less than to a specified value.
Search Less Than or Equal	Search for all array values less than or equal to a specified value.

Use the **Array Move (ARRAY_MOVE)** function to copy a specified number of data elements from a source array to a destination array.

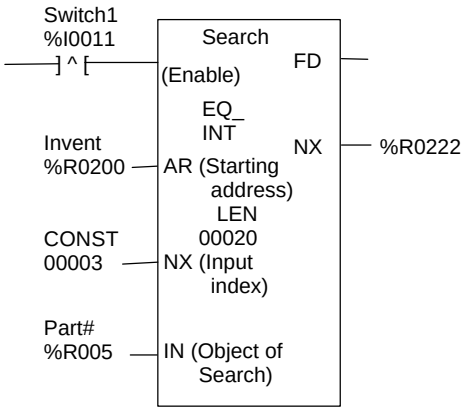
Table Function: ARRAY
(INT, DINT,BIT, BYTE, WORD)



	Description
Enable	When the function is enable, the operation is performed.
SR	SR contains the starting address of the source array. For ARRAY_MOVE_BIT, any reference may be used ; it does not need to be byte aligned. However, 16 bits, beginning with the reference address specified, are displayed on-line.
SNX	SNX contains the index of the source array
DNX	DNX contains the index of the destination array
N	N provides a count indicator
OK	The OK output is energized whenever enable is energized
DS	DS contains the starting address of the destination array. For ARRAY_MOVE_BIT, any reference may be used; it does not need to be byte aligned. However, 16 bits, beginning with the reference address specified, are displayed on-line.
LEN	LEN specifies the number of elements starting at SR and DS that make up each array.

Use the **Search Array** function to search for array values.

Table Function: Search Equal (INT, DINT,BYTE, WORD)

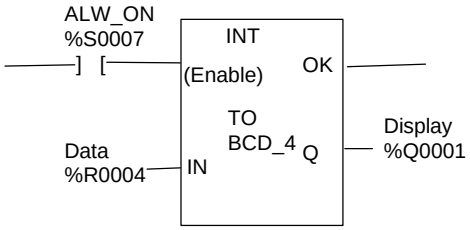


	Description
Enable	When the function is enabled,the operation is performed.
AR	AR contains the starting address of the array to be searched.
Input NX	Input NX contains the index into the array at which to begin the search.
IN	IN contains the object of the search.
Output NX	Output NX holds the position within the array of the search target.
FD	FD indicates that an array element has been found and the function was successful.
LEN	LEN specifies the number of elements starting at AR that make up the array to be searched. It may be 1 to 32,767 bytes or words.

Conversion

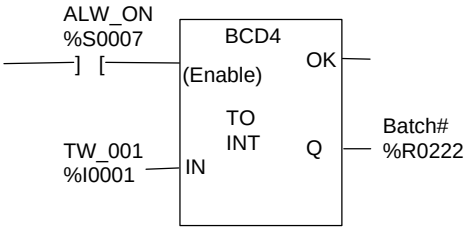
Conversion functions. To convert a data item from one number type to another. Many programming instructions, such as math functions, must be used with data of one type**.**

Conversion: Integer to BCD (INT)



	Description
Enable	When the function is enable, the conversion is performed.
IN	IN contains a reference for the integer value to be converted to BCD-4
OK	The OK output is energized when the function is performed without error.
Q	Output Q contains the BCD-4 form of the original value in IN.

Conversion: BCD to Integer (INT)



	Description
Enable	When the function is enabled, the version is performed.
IN	IN contains a reference for the BCD-4 value to be converted to integer.
OK	The OK output is energized whenever enable is energized, unless the data is out of range.
Q	Output Q contains the Integer form of the original value in IN.

Control

The following instructions may be used to limit program execution and alter the way the CPU executes application programs.

DO I/O — The DO I/O (DOIO) function is used to update inputs or outputs for one scan while the program is running. The DOIO function can also be used to update selected I/O during the program in addition to the normal I/O scan.

MCR — All rungs between an active Master Control Relay (MCR) and its corresponding. End Master Control Relay (ENDMCR) function are executed without power flow to coils. An ENDMCR function associated with the MCR is used to resume normal program execution.

Jump — Use the JUMP instruction to cause a portion of the program logic to be bypassed. Program execution will continue at the LABEL specified. When the JUMP is active, all coils within its scope are left at their previous states.
This includes coils associated with timers, counters, latches, and relays.

SVCREQ — Service Request - #13 Shut down PLC
#14 Clear fault table
#15 Read last fault
#16 Read elapsed time Clock

Unit Two – Configuration Lab

At the end of this lab you should be able to:

- Invoke and customize the VersaPro workbench environment
- Create a hardware configuration for the VersaMax CPU station
- Connect to a VersaMax PLC CPU and download a configuration

USING VERSAPRO

VersaPro provides a single programming interface to configure your PLC hardware, create and edit PLC logic, and monitor the execution of the PLC program.

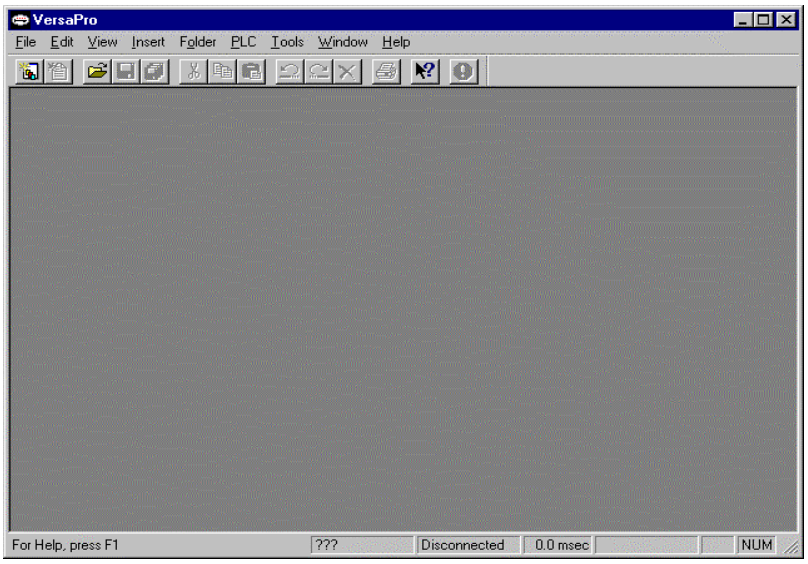
Component		Description
	Folder Browser	VersaPro is a flexible software package. You can select a default language and customize window and editor displays.
	Hardware Configuration	VersaPro's default hardware platform is the VersaMax PLC. You can change the hardware platform to create programs and hardware configuration for all Series 90-30 and VersaMax PLCs.
	Logic Editors	There are two editors for creating logic: the Instruction List Editor and the Ladder Diagram Editor. You can convert programs and blocks from IL to RLD, and visa-versa.
	Syntax Checking	VersaPro provides a tool to check that your program is syntactically valid before storing to the PLC.
	PLC Communications	VersaPro supports serial and Ethernet connections to the Series 90-30 PLCs, and serial connections to the VersaMax PLC.
	Online Monitoring	Online monitoring and control is supported in editors and Reference View Tables and Variable View Tables.
	Fault System	The fault system provides quick connection to view PLC and I/O fault tables in the PLC.
	Information Window	Displays the results of actions performed in VersaPro

The following abbreviations are used in VersaPro:

Abbreviation	Component
IL	Instruction List
RLD	Relay Ladder Diagram
RVT	Reference View Table
VDT	Variable Declaration Table
VVT	Variable View Table

THE WORKBENCH WINDOW

When you first start VersaPro, the workbench displays. You can perform the following tasks from the workbench:

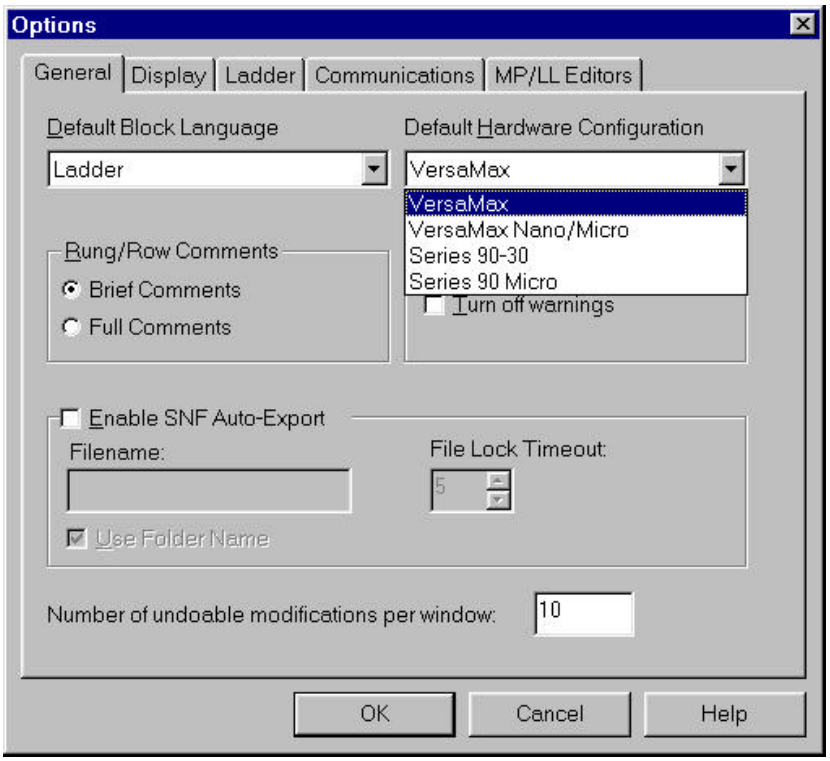


- Create and Open Folders
- Backup and Restore VersaPro folders
- Setup communications parameters, Connect to the PLC and view Fault Tables
- Customize Workbench Options

Setting Workbench Options

VersaPro allows you to customize editor and window options. It is important to confirm the default language and hardware configuration before creating a new folder. Other options may be customized before opening a folder, or after the folder is opened.

VersaPro's default hardware configuration is the VersaMax PLC. The default block language is Ladder Diagram.
To change the default configuration so that you can create programs and hardware configuration for the Series 90-30, or to customize display and editing options, select the Tools menu and choose Options. The Options dialog box will appear as above.



- *Setting Block Language*: The default language is Ladder Diagram. To change the default block language to Instruction List, select the General tab. Click the Default Block Language and select Instruction List

- *Changing Hardware Configuration default*: The default hardware configuration is VersaMax. To change the default hardware configuration, select the General tab and set the Default Hardware Configuration field.

- *Setting Full or Brief comments*: In the LD and IL Editors, comments can be configured to be viewed in full or brief mode. The default is to view comments in brief mode. To change the default, select the General Tab and check the Full Comments button.

- *Setting Syntax Checking (Turn off Warnings):* VersaPro performs syntax checking on logic – if logic is not syntactically correct, warnings are displayed in the information window. To suppress the warnings, select the General tab and check the Turn Off Warnings box.

- *Setting the number of undoable actions:* VersaPro allows you to configure the number of undos/redos available during editing sessions. The default number is 10. The valid range is 1 to 100.

Setting Display Options

VersaPro allows you to set text font, size and color separately.
Perform these steps to set display options:

1. Choose Options from the **TOOLS** menu, or from the LD or IL Editor CSM.
2. The **OPTIONS** dialog box appears.
3. Select the Display tab.
4. From the Category list, select the **EDITOR** or **TABLE** whose display options you wish to configure.
5. The Colors list displays the items you can configure for the selected editor or table.
6. Select the color to be used for each configurable area of the selected editor or table. How many areas are configurable, and what those areas are, differs based on the window whose display you are configuring.

   Note: Certain combinations of colors and backgrounds can be difficult to read.

7. Select the font type and font size to be used to display text in the selected editor or table. Click **OK**

   Note: You cannot change the font used for Column and Row headings.

   The selected colors are applied to selected areas of the selected editor or table. The selected font is applied to all text within the selected editor or table. If the selected editor or table is visible, it appears with its changed colors, font and size.

   Note: Selected fonts are used when you print your program.

CREATING A NEW FOLDER

A folder contains all of the components required to program, configure and monitor your VersaMax and Series 90-30 PLCs. The components include Hardware Configuration, Variable Declaration Table, View Tables and program blocks. You can also insert other windows-components into the folder, like user documents and spreadsheets.

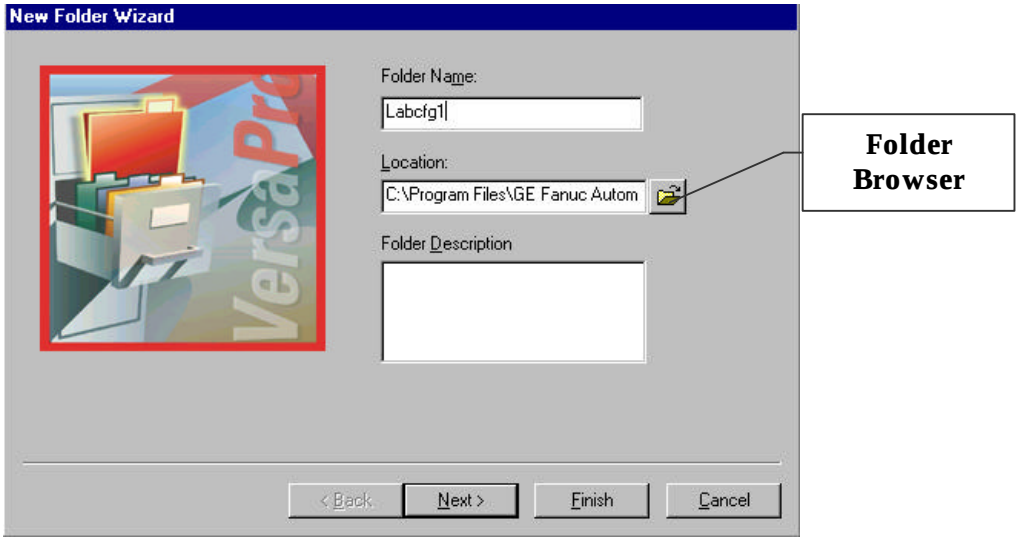
VersaPro allows you to have only one folder open at a time. However, you can have multiple instances of VersaPro running on your PC.

VersaPro provides several options when creating a new folder. You can create an empty folder, or create a new folder based on an existing VersaPro folder. VersaPro also provides an option for you to create a folder by importing content from Logicmaster™ 90 or Control for use with a 90-30 Series or a VersaMax PLC.

To create a new folder

The default language for new folders is Ladder Diagram. The default hardware configuration is VersaMax. If you want to change the default language or target hardware for your new folder, select the Tools menu and choose Options. Change the settings in the Options dialog box.
To create a new, empty folder from the VersaPro workbench:

1. Click the **NEW FOLDER** button , choose New Folder from the File menu, or press Ctrl+N. The New Folder Wizard will appear.

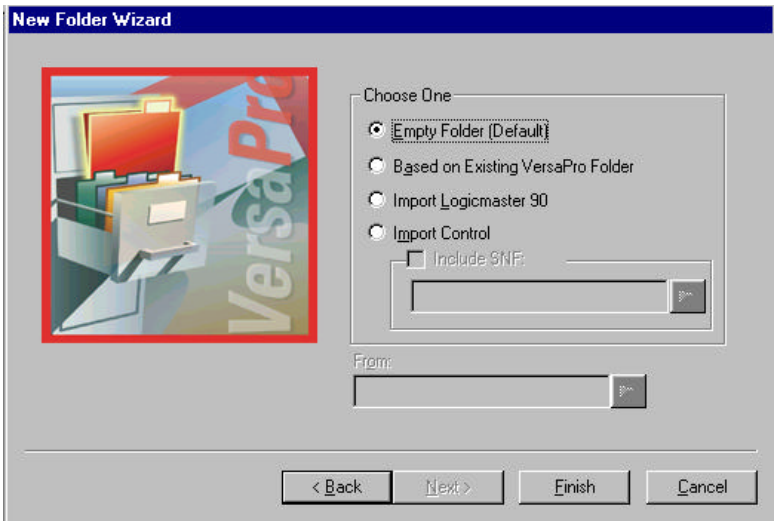


2. In the **FOLDER NAME FIELD**, type a name for the new folder. A folder name is required.

3. If you want to save the new folder to a location different from the default, click on the **BROWSE** button opposite the Location field.

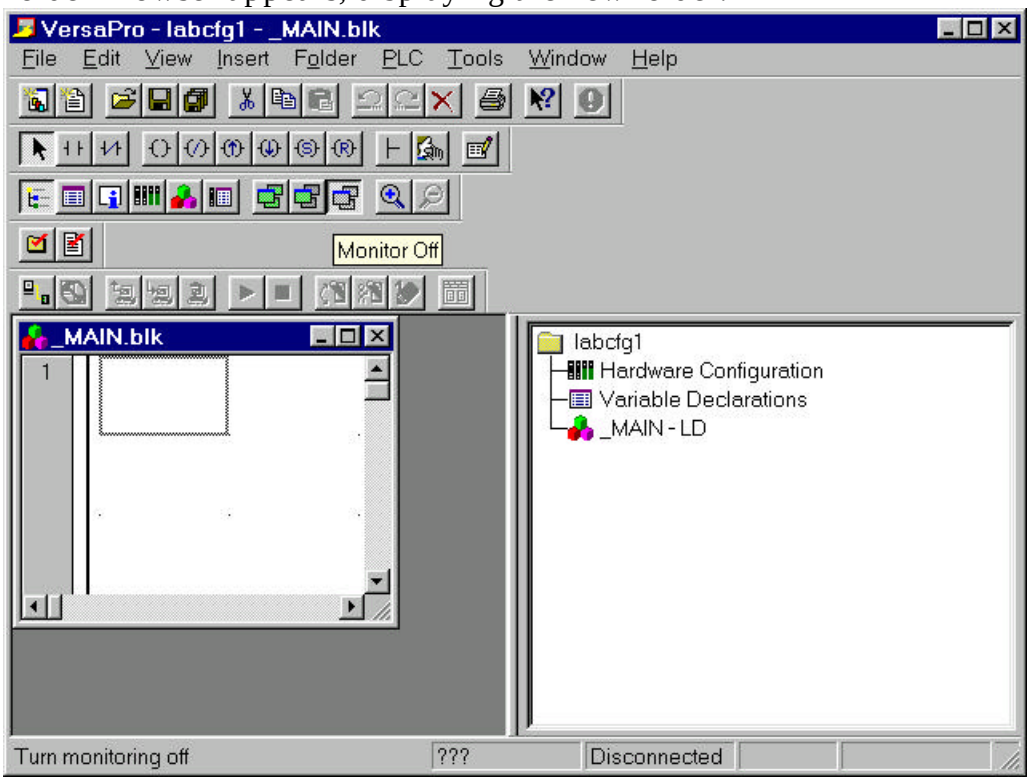
4. Enter a folder description in the Folder Description field. You may enter up to 64 characters. This field is optional.

5. Click the **NEXT** button. The next screen of the New Folder Wizard will appear.



6. The Empty Folder (Default) button is automatically selected. Click **FINISH** to create the new, empty folder.

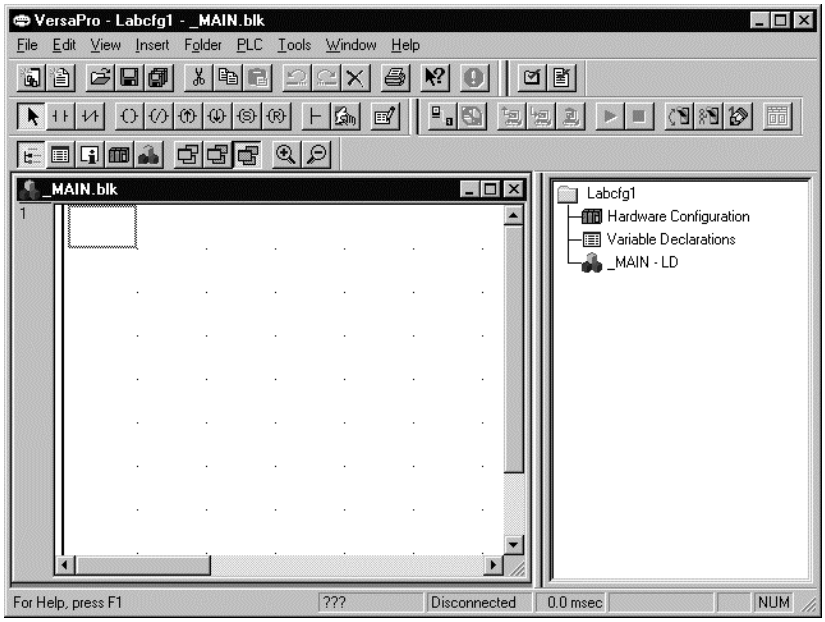
The Folder Browser appears, displaying the new folder.



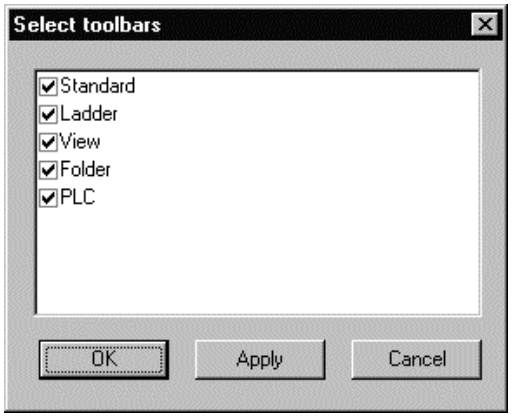
The size and position of the Folder Browser depend on its retentive properties. The _MAIN block will appear in the default language's editor.

CUSTOMIZING THE WORKSPACE

The screen below is the MAIN LADDER VersaPro workspace, which you can customize in many different ways. All settings are saved in the folder on an "application" basis.



1. To custom-select the toolbars, first select **VIEW**, then **TOOLBARS** on the Menu bar.
2. Select each of the toolbars by clicking on them, then click **OK**.
3. To move the toolbars around, click and hold on an area between the toolbar buttons and drag it to where you would like it positioned on the workspace. Do the same for the other toolbars.



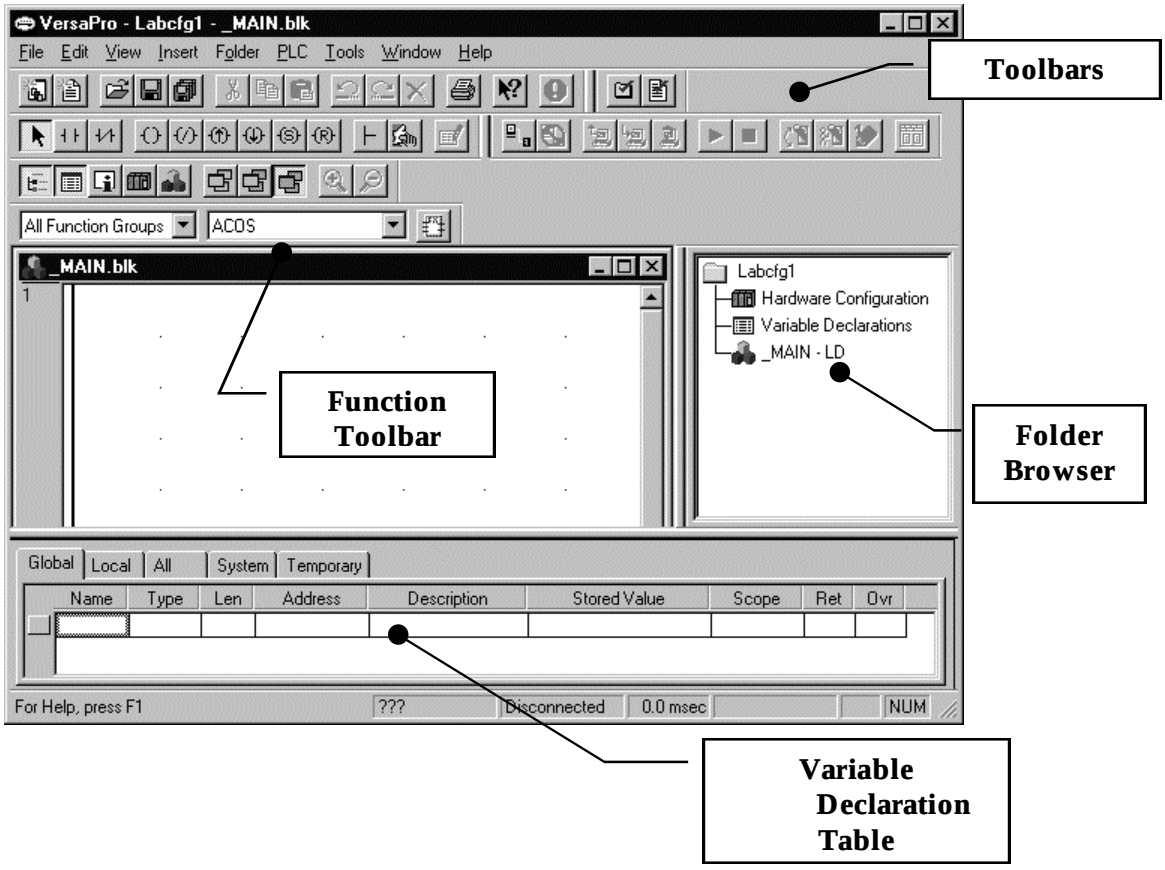
4. Set up the Function Toolbar by selecting **VIEW** on the Menu bar, then **FUNCTION TOOLBARS**. Click on either the **EXPANDED** or **COMPACT** selection and note how each appears on the workspace.

5. The Compact Toolbar can be placed anywhere on the workspace. Click, hold and drag it to wherever you would like to have it on the workspace.

6. Check to see if the Variable Declaration Table and Folder Browser are turned on. If not, select **VIEW** on the Menu bar, then toggle the **FOLDER BROWSER** and **VARIABLE DECLARATION TABLE** buttons.

7. Place your cursor on the edge of the Folder Browser and drag it to the position on the workspace shown below. Repeat this process, if necessary, with the Variable Declaration Table and Function Toolbar.

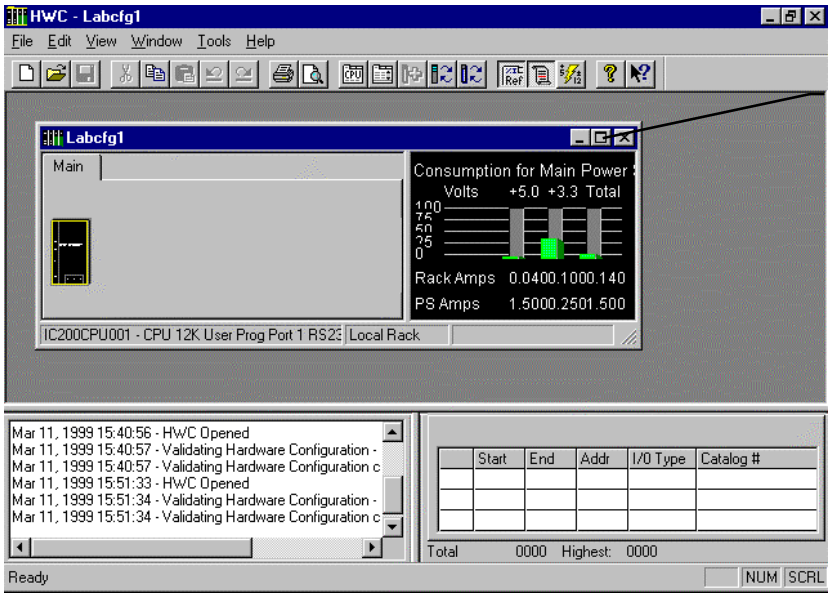
8. Move the mouse cursor until the double arrows (↔) appear and resize the boxes for the best fit. Your workspace should look similar to the one below.



VERSAMAX CONFIGURATION

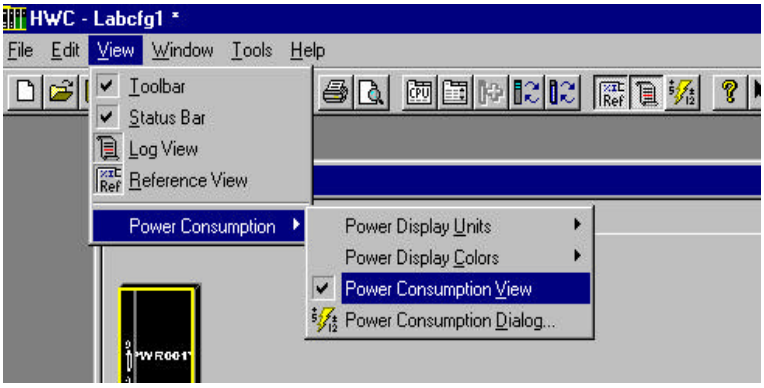
The rack system for the configured CPU hardware will display, along with a graph showing power consumption statistics for the module, a table showing references used in hardware configuration and a log which maintains a record of HWC events

1. Open the hardware configuration window by clicking on the **VIEW HARDWARE CONFIGURATION** button on the toolbar or by double clicking on **Hardware Configuration** in the Folder Browser.
2. This screen shows the default Hardware Configuration for a VersaMax system.
3. Maximize this workspace by clicking on the "Maximize" icon in the upper right corner of the workspace.

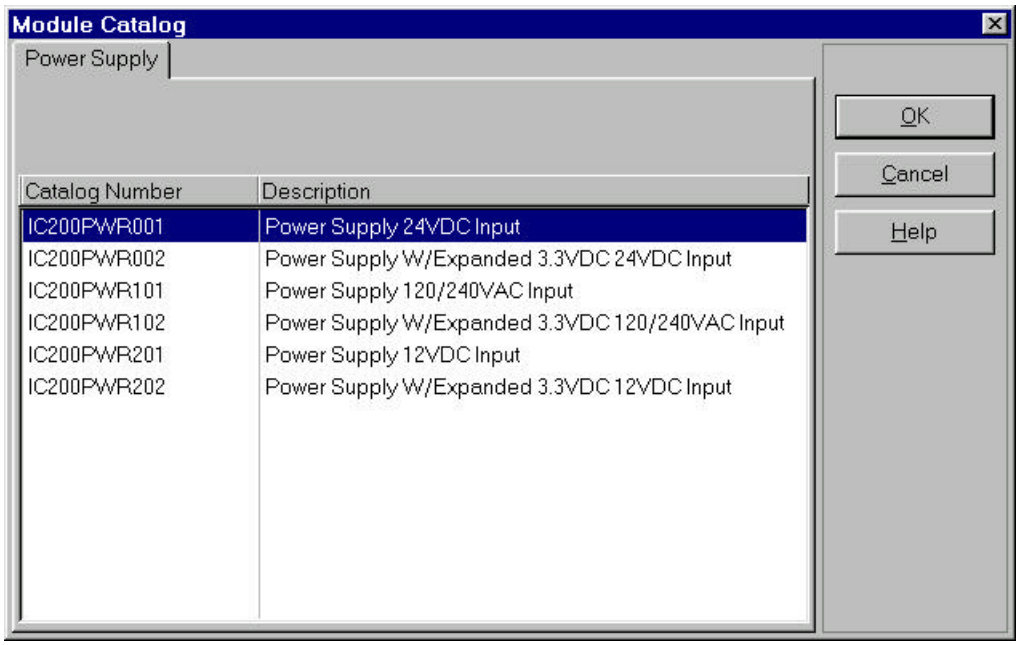

HWC - Labcfg1
File Edit View Window Tools Help
Labcfg1
Main
Consumption for Main Power
Volts +5.0 +3.3 Total
Rack Amps 0.0400.1000.140
PS Amps 1.5000.2501.500
IC200CPU001 - CPU 12K User Prog Port 1 RS23
Local Rack
Mar 11, 1999 15:40:56 - HWC Opened
Mar 11, 1999 15:40:57 - Validating Hardware Configuration -
Mar 11, 1999 15:40:57 - Validating Hardware Configuration c
Mar 11, 1999 15:51:33 - HWC Opened
Mar 11, 1999 15:51:34 - Validating Hardware Configuration -
Mar 11, 1999 15:51:34 - Validating Hardware Configuration c
Start End Addr I/O Type Catalog #
Total 0000 Highest: 0000
Ready
NUM SCRL

Maximize

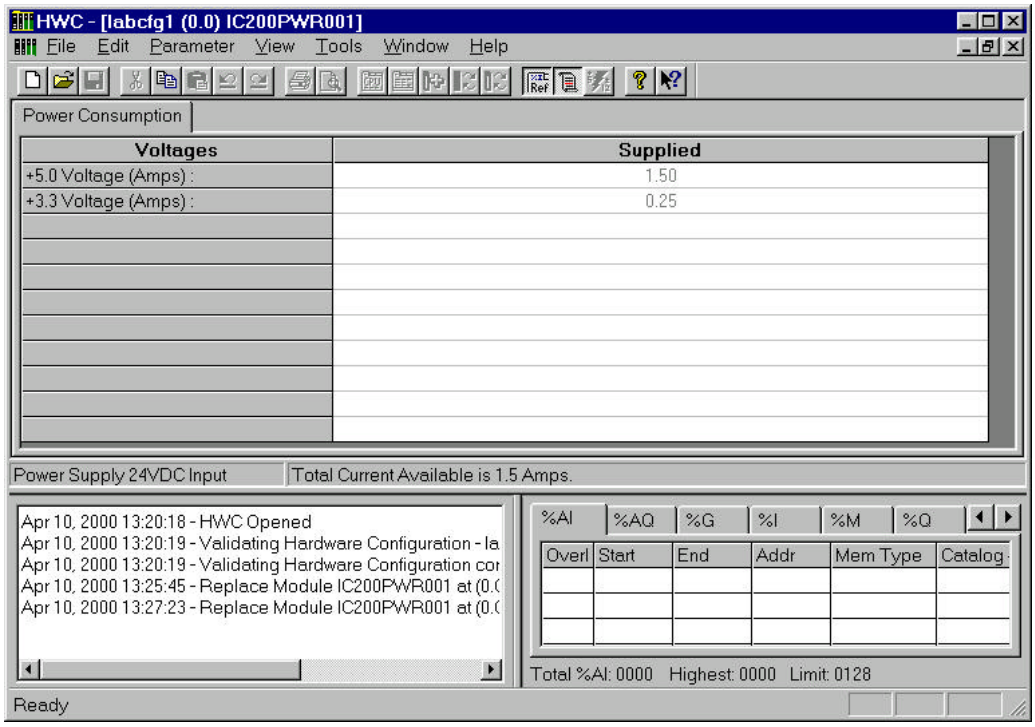
4. You may also optionally turn off the "Power Consumption" window from the toolbar menu by clicking on the **VIEW POWER CONSUMPTION** selection and clicking on **POWER CONSUMPTION VIEW** to the right.



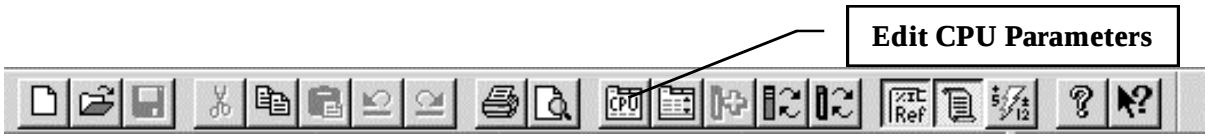
5. Right mouse click on the Power Supply slot. If the power supply type is incorrect, go to **REPLACE MODULE** and select the proper power supply from the pop-up list. Click **OK** to replace the module.



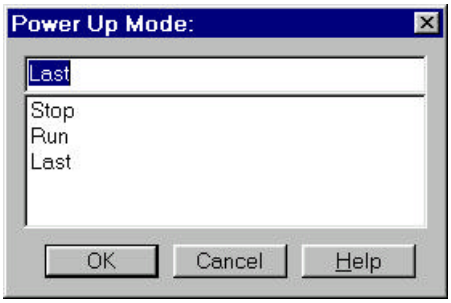
At this point, the "Power Consumption" window appears. Note that the window shows the power consumption of the installed modules.



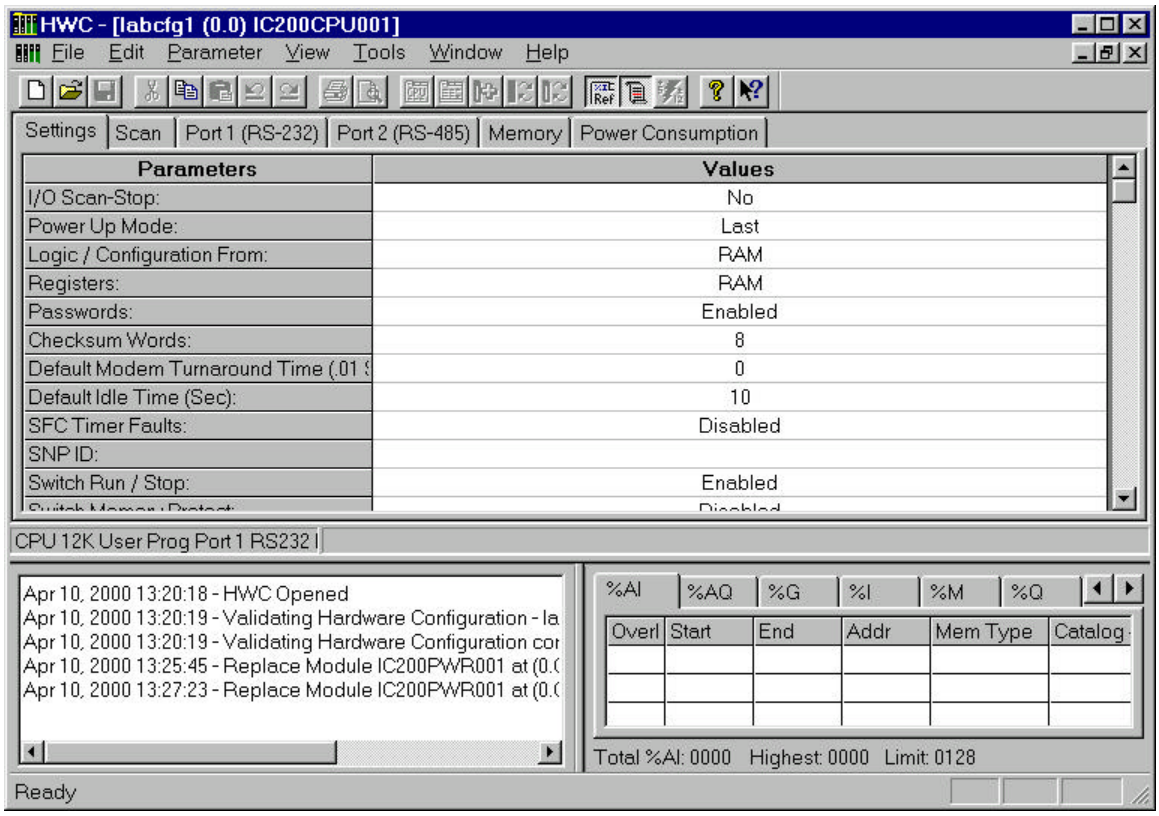
6. To configure the CPU Parameters right mouse click on the power supply and go to **CONFIGURE CPU PARAMETERS** or by selecting the **EDIT CPU PARAMETERS** button on the toolbar.



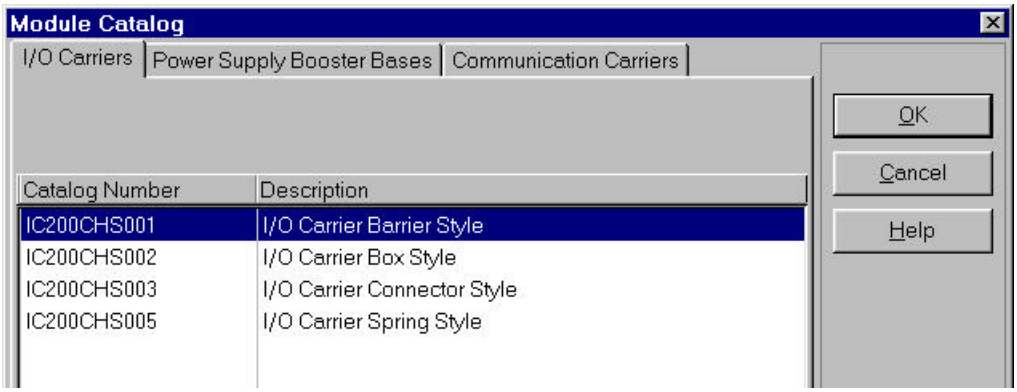
7. Double click on each value to see a pop-up menu of the configuration value choices.

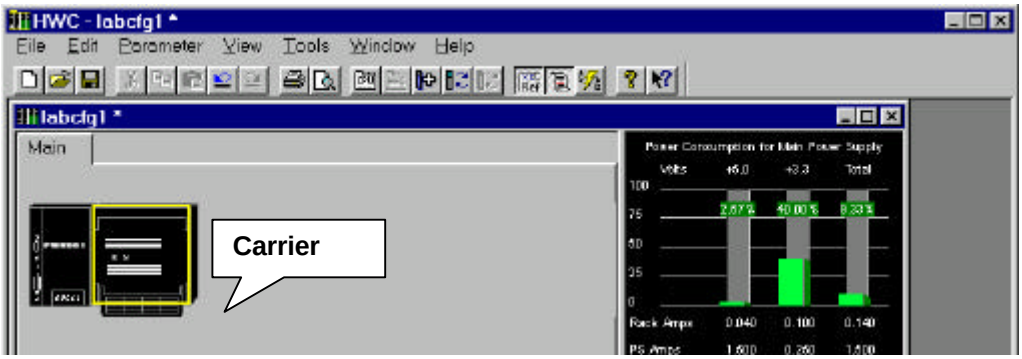


8. Examine the other CPU parameters by clicking on the **Scan**, **Port(x)**, **Memory** and **Power Consumption** tabs. Note that some selections may be fixed for the CPU.

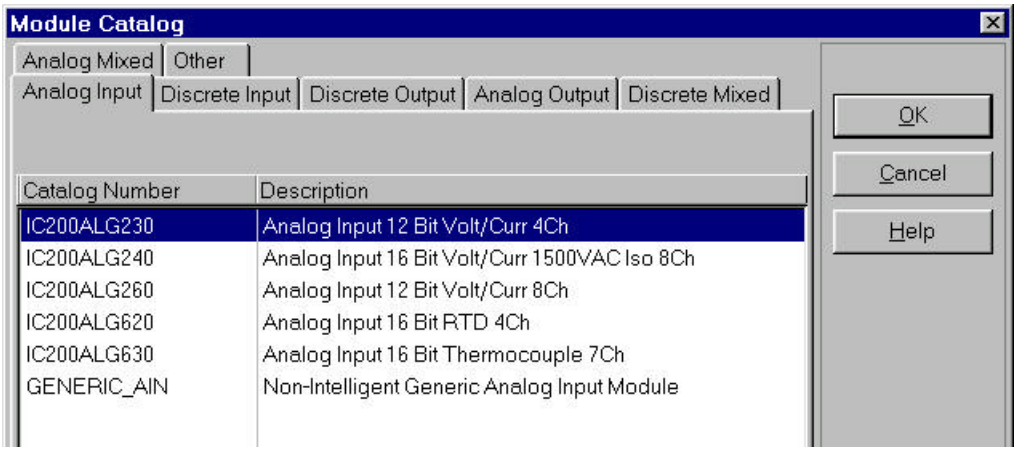

HWC - [labcfg1 (0.0) IC200CPU001]
File Edit Parameter View Tools Window Help
Settings Scan Port 1 (RS-232) Port 2 (RS-485) Memory Power Consumption
Parameters
Values
I/O Scan-Stop: No
Power Up Mode: Last
Logic / Configuration From: RAM
Registers: RAM
Passwords: Enabled
Checksum Words: 8
Default Modem Turnaround Time (.01 S 0
Default Idle Time (Sec): 10
SFC Timer Faults: Disabled
SNP ID:
Switch Run / Stop: Enabled
CPU 12K User Prog Port 1 RS232
Apr 10, 2000 13:20:18 - HWC Opened
Apr 10, 2000 13:20:19 - Validating Hardware Configuration - la
Apr 10, 2000 13:20:19 - Validating Hardware Configuration cor
Apr 10, 2000 13:25:45 - Replace Module IC200PWR001 at (0.(
Apr 10, 2000 13:27:23 - Replace Module IC200PWR001 at (0.(
%AI %AQ %G %I %M %Q
Overl Start End Addr Mem Type Catalog
Total %AI: 0000 Highest: 0000 Limit: 0128
Ready

9. Add a carrier for the next module by right-clicking on the power supply and selecting “Add Carrier / Base”.

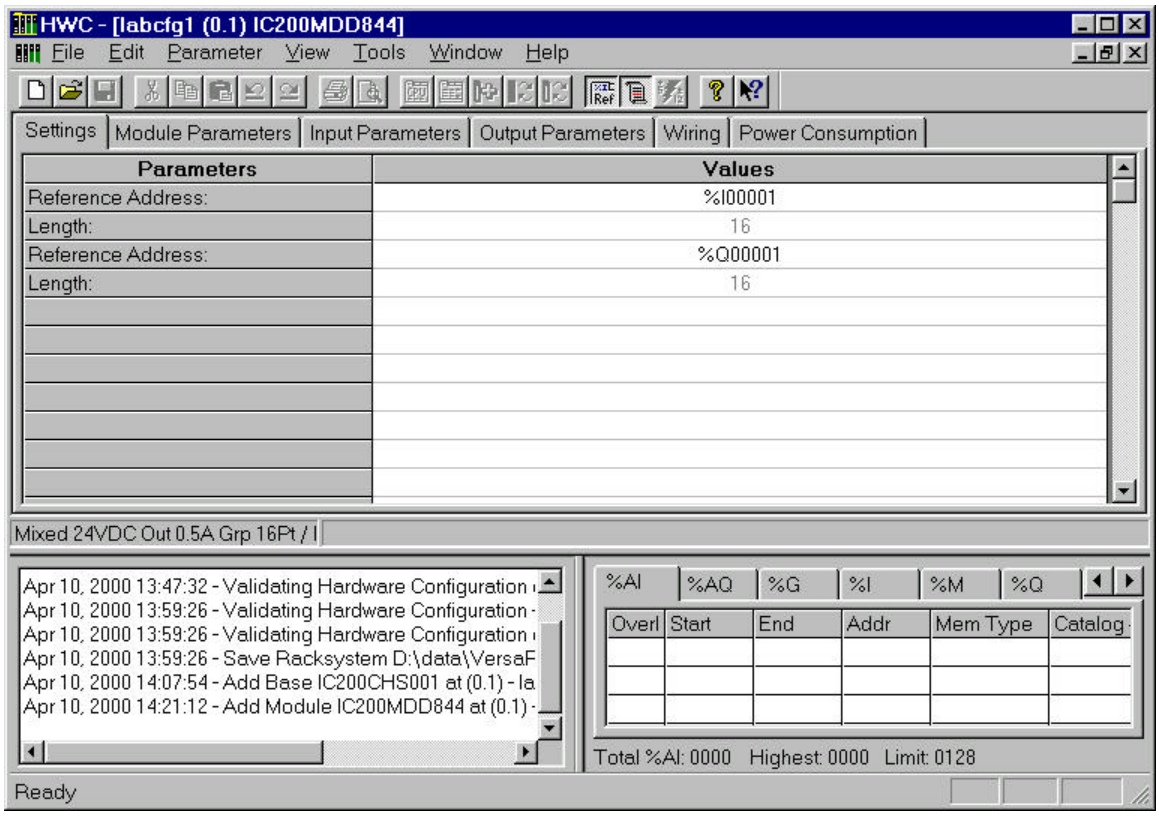




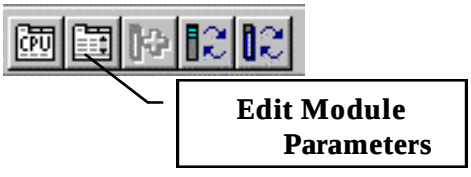
10. Add the next module by right-clicking on a carrier and selecting **ADD MODULE**. A “Module Catalog” window will appear, with tabs for each of the VersaMax module types.



11. Select the module type from the tab selection. Highlight the proper module in the list and click **OK** to enter the module "Parameter" window.

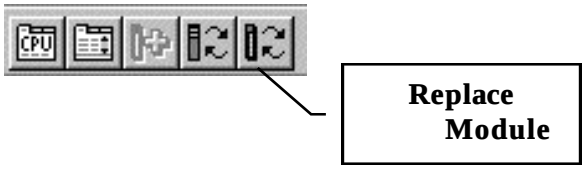


12. Click on each parameter tab and examine the selections for the module under "Values". Double click on each value to see a pop-up menu of the configuration value choices.
13. Select the **Settings** tab. Verify or change the module's **Reference Address** as well as any other parameters.
14. By selecting the **Wiring** tab, you may enter a description of each I/O point under the "Wiring Information" heading.
15. Continue adding the remaining modules in the same manner, clicking **HELP** for additional configuration information.
16. You may modify any existing module's configuration by right clicking on it and selecting **CONFIGURE PARAMETERS** or by selecting the **EDIT MODULE PARAMETERS** button on the toolbar.

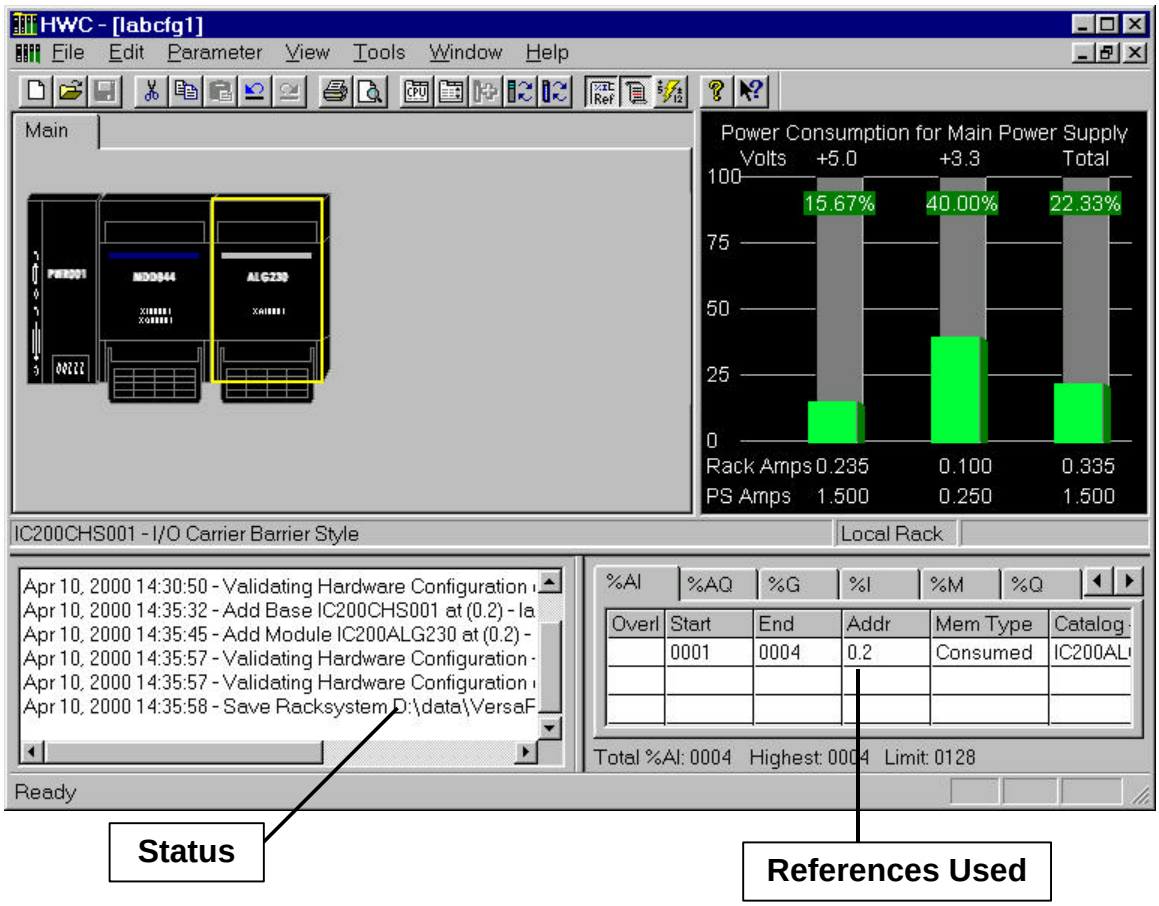


17. A module may be deleted by right-clicking on it and selecting **DELETE MODULE** or by selecting **EDIT,** then **DELETE** on the Menu bar.

18. A module can be replaced by right clicking on it and selecting **REPLACE MODULE** or by selecting the **REPLACE MODULE** button on the toolbar.



19. Click the **SAVE** button on the toolbar to store the newly created configuration to the lab folder.

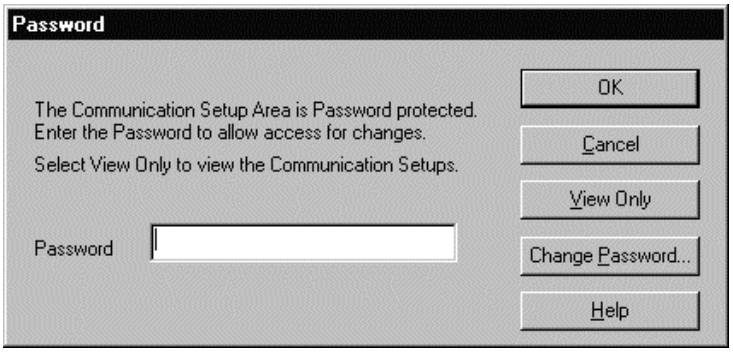

HWC - [labcfg1]
File Edit Parameter View Tools Window Help
Main
Power Consumption for Main Power Supply
Volts +5.0 +3.3 Total
100
75
50
25
0
15.67%
40.00%
22.33%
Rack Amps 0.235 0.100 0.335
PS Amps 1.500 0.250 1.500
IC200CHS001 - I/O Carrier Barrier Style
Local Rack
Apr 10, 2000 14:30:50 - Validating Hardware Configuration
Apr 10, 2000 14:35:32 - Add Base IC200CHS001 at (0.2) - la
Apr 10, 2000 14:35:45 - Add Module IC200ALG230 at (0.2) -
Apr 10, 2000 14:35:57 - Validating Hardware Configuration
Apr 10, 2000 14:35:57 - Validating Hardware Configuration
Apr 10, 2000 14:35:58 - Save Racksystem D:\data\VersaF
%AI %AQ %G %I %M %Q
Overl Start End Addr Mem Type Catalog
0001 0004 0.2 Consumed IC200AL
Total %AI: 0004 Highest: 0004 Limit: 0128
Ready
Status
References Used

1. Note that the “Status” window keeps a running tally of the operations you have performed and the ´’References Used’’ window allows you to look at the I/O references you have assigned in configuration.
2. Click on the %AI, %I and %Q tabs to look at each of the reference I/O assignments.
3. When finished, close the “Hardware Configuration” window by clicking on the “X” in the upper right corner of the window. This will return you to the MAIN LADDER VersaPro Workspace.

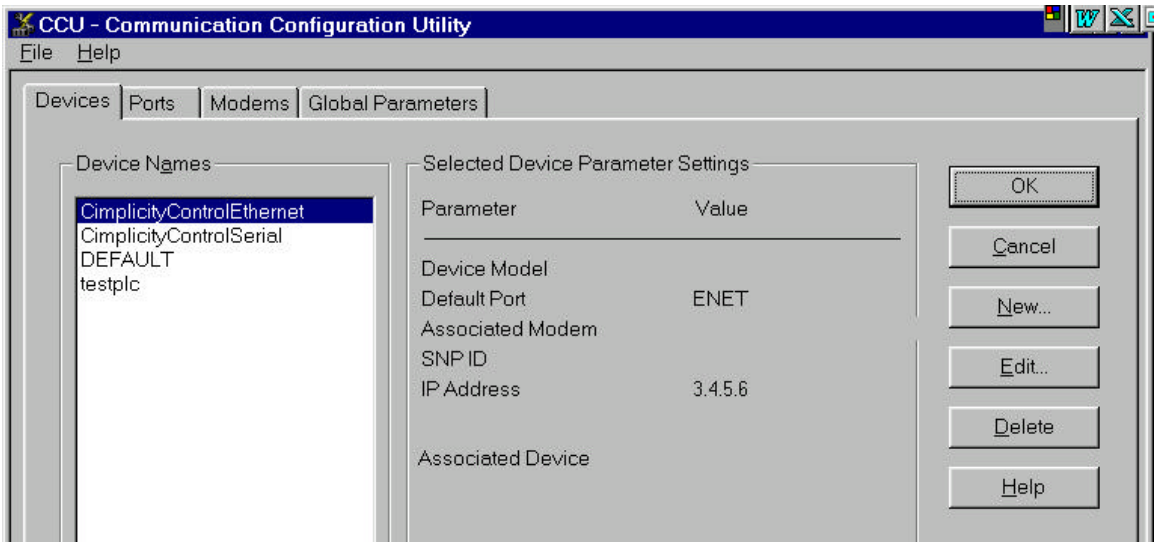
STORING THE CONFIGURATION TO THE VERSAMAX PLC

You can connect to a PLC from VersaPro over a serial or Ethernet connection, depending on the capabilities of the PLC. Connections are made by identifying a device and selecting a port. The device specifies information necessary to successfully connect including the default PC communication port, target PLC type, PLC SNP address and PLC IP address. The port identifies the PC port used to initiate communications to the PLC.

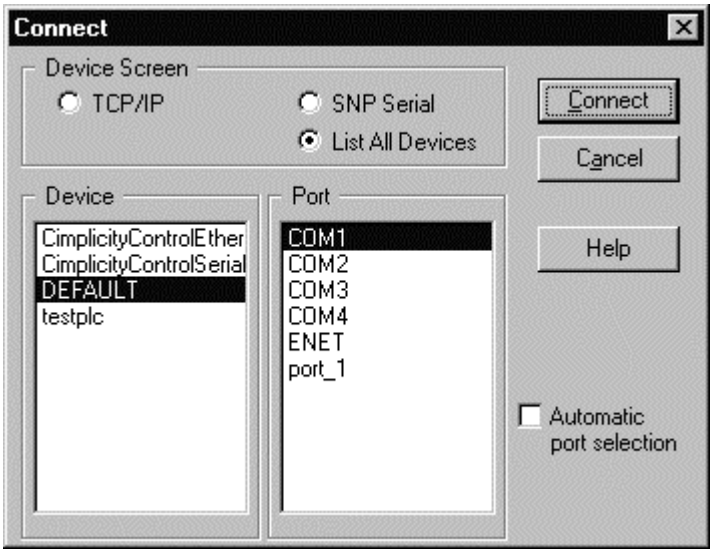
1. On the Menu bar, click on **TOOLS,** then **COMMUNICATIONS SETUP.** A “Password” window will appear. Enter the correct password (netutil) or select **VIEW ONLY** to enter the “Communications Configuration Utility” window.



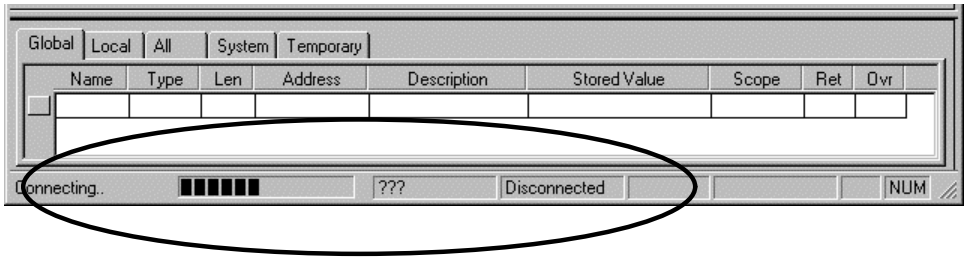
2. By selecting **NEW** from the Communication Configuration Utility, you can create a new “Device Name” or **EDIT** an existing one. Select the “Devices”, “Ports” and “Modems” tabs if required. Click on **OK** to complete the operation.

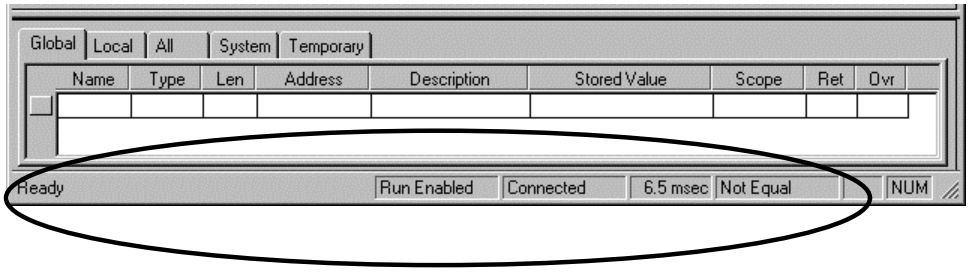


3. Select the **CONNECT** button on the tool bar or on the Menu bar, click on **PLC**, then **CONNECT**. A pop-up window will appear with **Device Screen**, **Device** and **Port** selections. Click on the **LIST ALL DEVICES** radio button and select the proper "Device" and "Port" for communications to the PLC.

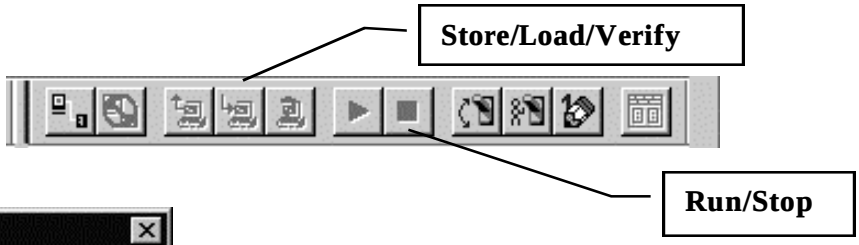


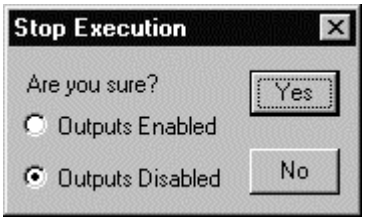
4. Click on the **CONNECT button** and the VersaPro software will attempt to communicate with the PLC. Note the Status information at the bottom of the screen which will show the "connect" status and inform you when the operation is complete. If the connection is successful, a "Ready" message along with the CPU Run Status, Scan time and whether or not the program is equal to the PLC information will be displayed.





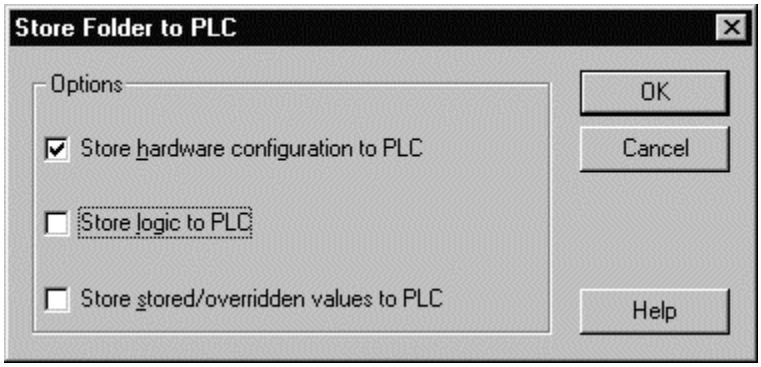
5. Should an error occur, a pop-up will appear with an indication of the problem. Insure that the cabling is correct and that the PLC is using the same baud rate specified when the CPU was configured.





6. If the CPU is in the RUN mode, select the **STOP PLC** button on the toolbar or select **PLC**, then **STOP** on the menu bar. Click on either "Outputs Enabled or Disabled" and select **YES** to stop the PLC.

7. Select the **STORE TO PLC** button on the toolbar or select **PLC**, then **STORE** on the menu bar.



8. Insure that only the **STORE HARDWARE CONFIGURATION** radio button is checked and select **OK** to begin the store operation.

9. This message will appear when the store operation is complete.

Unit Three - Programming Lab

At the end of this lab you should be able to:

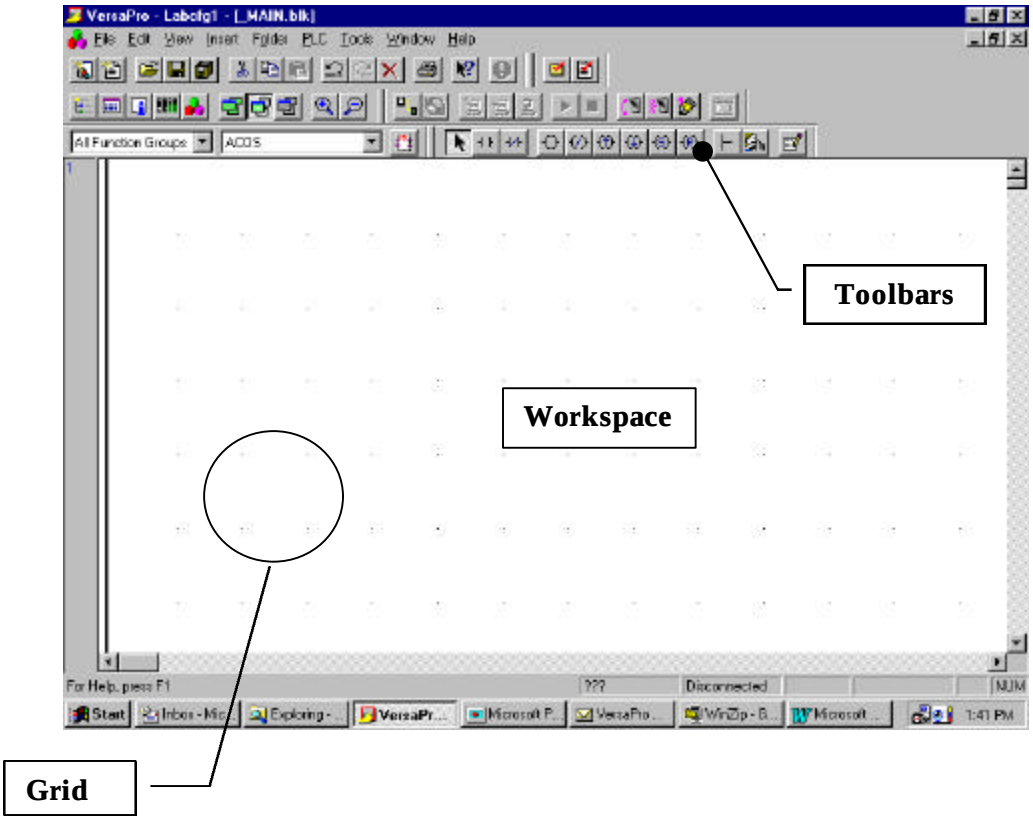
- Create a Program Folder using VersaPro
- Create a Ladder or Instruction List Program using the VersaPro Editor
- Compile and download your Program to a VersaMax PLC CPU
- Monitor your Program and Reference Tables at runtime
- View different Status Tables within the VersaMax PLC
- Insert objects such as EXCEL, or WORD files within your folder
- Print your program and configuration from VersaPro

GETTING STARTED

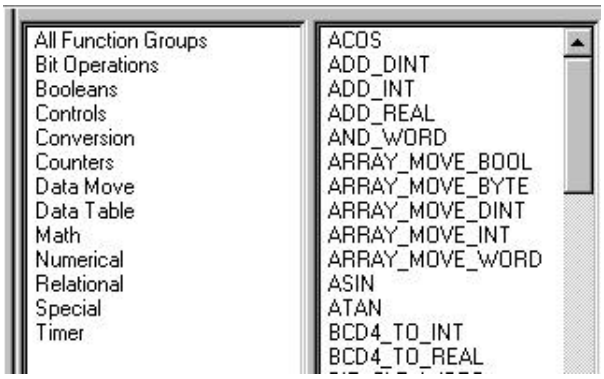
The Ladder Diagram Editor is used to create programs using the Ladder Diagram programming language. Ladder Diagram is used to graphically represent the actions to be performed on a PLC.

The editor window is divided into rows (rungs). Each rung is divided into cells. Each cell can accept a logic element (function block or relay function.). The LD Editor allows you to enter ladder logic elements via keyboard, mouse, or toolbar. All elements are inserted by overwriting the contents of the selected cell.

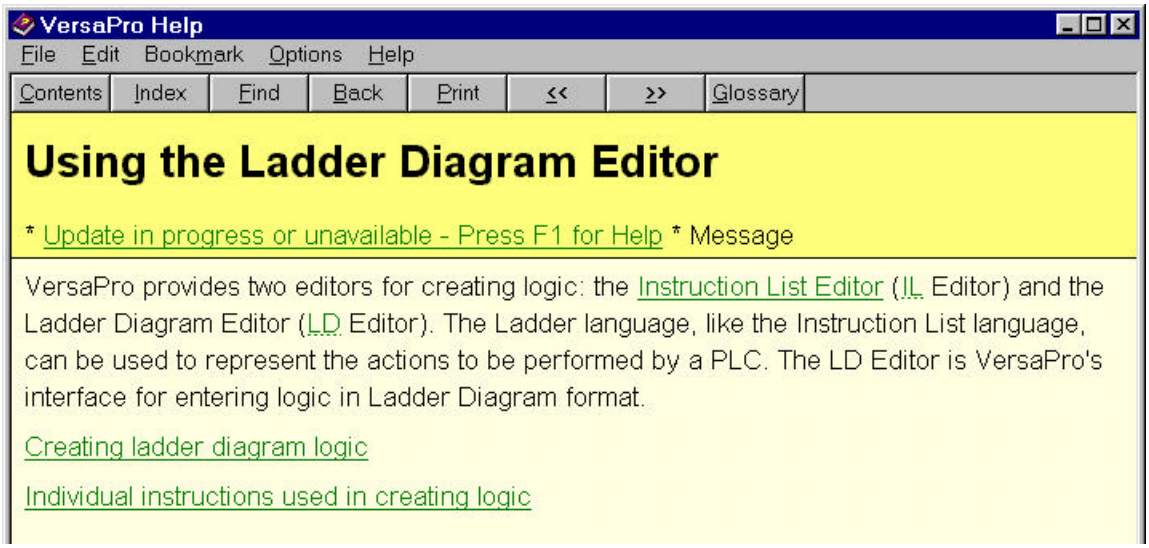
The screen below is the MAIN LADDER VersaPro workspace, which you can customize in many different ways. All settings are saved in the folder on an "application" basis.



1. Set up the Function Toolbar by selecting **VIEW** on the Menu bar, then **FUNCTION TOOLBARS**. Click on either the **EXPANDED** or **COMPACT** selection and note how each appears on the workspace.



2. The Compact Toolbar can be placed anywhere on the workspace. Click, hold and drag it to wherever you would like to have it on the workspace.
3. Move your mouse cursor to each of the menu bar buttons. A short description of each button will appear below the cursor.
4. Help is available by mouse clicking in a workspace area and pressing function key **F1.**

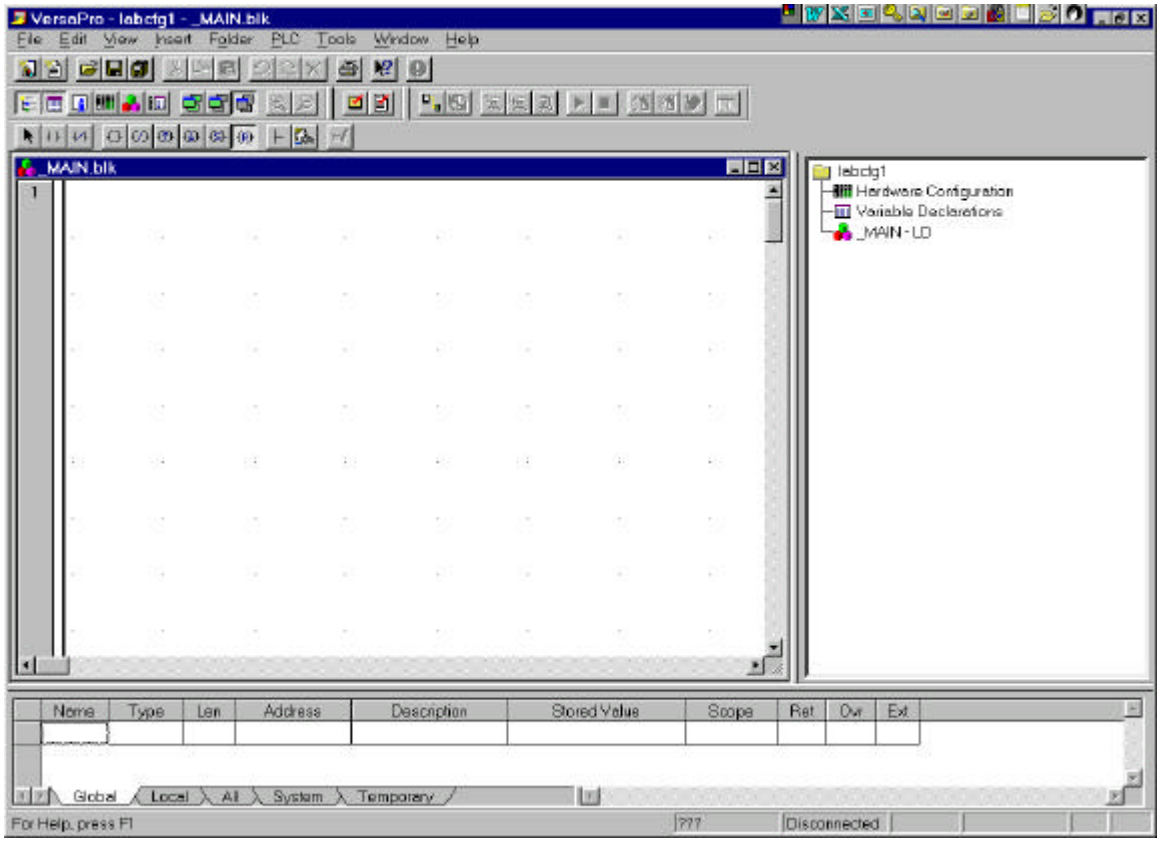

VersaPro Help
File Edit Bookmark Options Help
Contents Index Find Back Print << >> Glossary
Using the Ladder Diagram Editor
* Update in progress or unavailable - Press F1 for Help * Message
VersaPro provides two editors for creating logic: the Instruction List Editor (IL Editor) and the Ladder Diagram Editor (LD Editor). The Ladder language, like the Instruction List language, can be used to represent the actions to be performed by a PLC. The LD Editor is VersaPro's interface for entering logic in Ladder Diagram format.
Creating ladder diagram logic
Individual instructions used in creating logic

5. You may also click on the toolbar "Help" button. This will add a question mark to the cursor and allow you to click on any workspace for help.

6. Try some of the other toolbar buttons to see what their functions are.

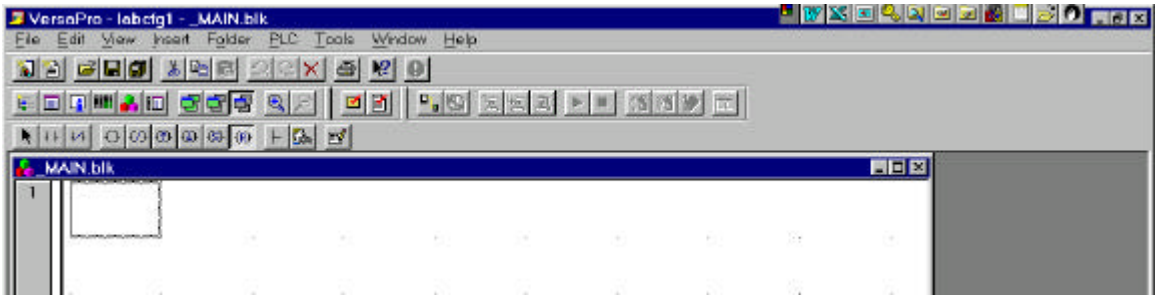
7. Check to see if the Variable Declaration Table, Information Window and Folder Browser are turned on. If not, turn them on using the proper button on the toolbar or select **VIEW** on the Menu bar, then toggle the **FOLDER BROWSER INFORMATION WINDOW** and **VARIABLE DECLARATION TABLE** buttons.

8. Place your cursor on the edge of the Folder Browser and drag it to the position on the workspace shown on the previous page. Repeat this process, if necessary, with the Information Window and Variable Declaration Table.


VersaPro - labcfg1 - _MAIN.blk
File Edit View Insert Folder PLC Tools Window Help
_MAIN.blk
labcfg1
Hardware Configuration
Variable Declarations
_MAIN - LD
Name Type Len Address Description Stored Value Scope Ret Ovr Ext
Global Local All System Temporary
For Help, press F1
Disconnected

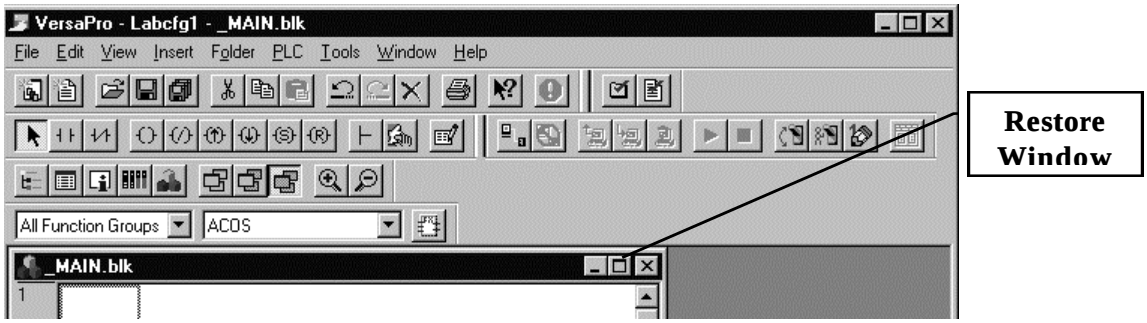
9. Move the mouse cursor until the double arrows (↔) appear and resize the workspaces for the best fit.

10. You may turn off the **FOLDER BROWSER**, **INFORMATION WINDOW** and **VARIABLE DECLARATION TABLE** for more room on the workspace by following the same procedure that was outlined in step #10.



ENTERING RELAY LADDER LOGIC

1. With the Ladder Editor workspace displayed, maximize it by clicking on the "Restore Window" button at the top right of the workspace.



2. You are now going to create a Relay Ladder Diagram so follow the next steps carefully. All of the function buttons will display their functions when the cursor is placed on them.

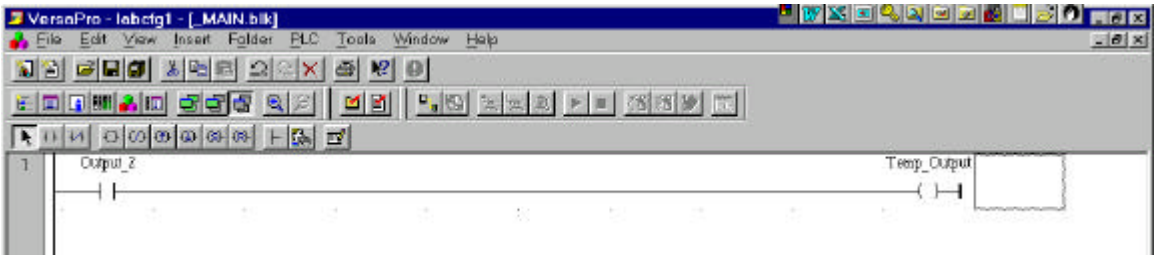
3. Move up to the Ladder toolbar and click on a NORMALLY OPEN CONTACT.

4. Place the mouse cursor in the upper left corner of the workspace and notice that an image of the contact appears to the right of the cursor. Click the mouse button again to "drop" the element on the workspace.

5. Return to the toolbar and click on the NORMAL POINTER button to remove the contact from the cursor.

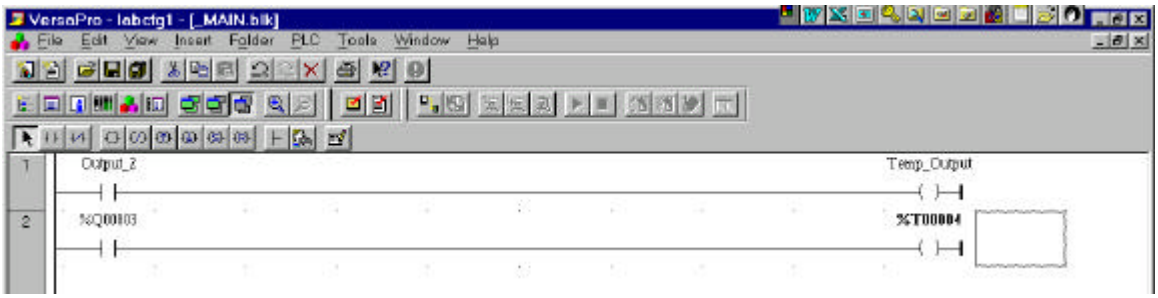
6. Another method to entering ladder logic elements is to double click on the area on the workspace that you wish to enter an element. A "pull down" box will appear with a list of all the logic function mnemonics.

7. Select a NOCON, click below the box and the element will be entered.
8. Double click on the element (in normal pointer mode) and a “pull-down” box will appear. Type 2Q in the box and press ENTER on the keyboard to enter the contact reference.
9. If you make a mistake, select the element by clicking on it and click on the DELETE SELECTION button on the toolbar or press the DELETE key on the keyboard to remove it.
10. Multiple items may be removed by clicking the mouse button (in normal pointer mode) and holding it while you select the elements you wish to remove.
11. An element may be moved on the workspace by clicking and holding the mouse button while you “drag” it to the new position. Release the button to “drop” the item in its new position.
12. Select the HORIZONTAL / VERTICAL WIRE button and move back to the right of the Output_2 normally open contact.
13. Note that a horizontal or vertical jumper will appear next to the cursor, depending on where it is placed.
14. Go back to the toolbar and select a Normally Open Coil. Click and drop the coil on the extreme top right side of the workspace.
15. Again, double click on the element and type Temp_Output in the “pull-down” box. Press the ENTER key to enter the reference.



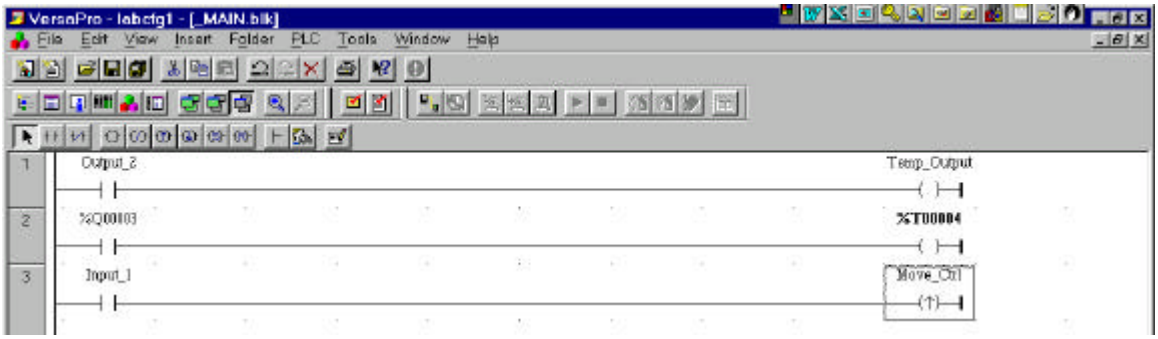
16. Return to the toolbar and select the NORMALLY OPEN CONTACT again and place a contact on the line below the rung you just completed. Enter %Q3 (3Q) as its reference.

17. Select a Normally Open Coil. Click and drop the coil on the extreme top right side of the workspace. Double click on the element and type %T4. Press the ENTER key to enter the reference.



18. Enter a NORMALLY OPEN CONTACT on the next rung. Enter Input_1 as its reference.

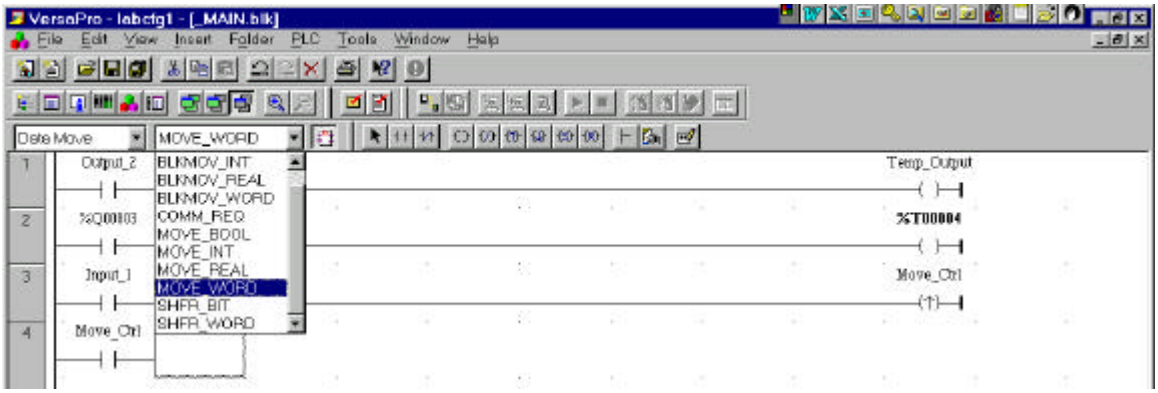
19. Enter a POSITIVE TRANSITION COIL at the right side of the rung with Move_Ctrl as its reference



20. Enter a NORMALLY OPEN CONTACT below the Input_1 contact. Double click on the contact (in normal pointer mode) and when the "pull-down" box appears, click on the arrow to the right of the box and note that all of the references previously entered are shown in a list.

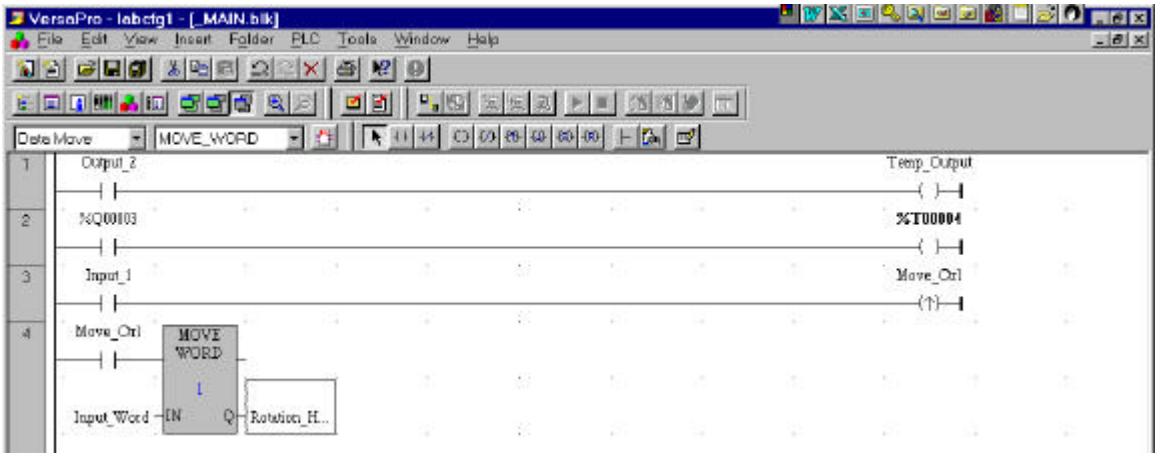
21. Select Move_Ctrl from the list and click below the box to enter the reference.

22. Enter a Data Move to the right of this contact by clicking on the arrow to the right of the Function Group toolbar and selecting DATA MOVE. Do the same for the Function Name toolbar and select MOVE_WORD.



23. Double click or enter on the MOVE_WORD function to display the function properties. Type 1 in the length box.

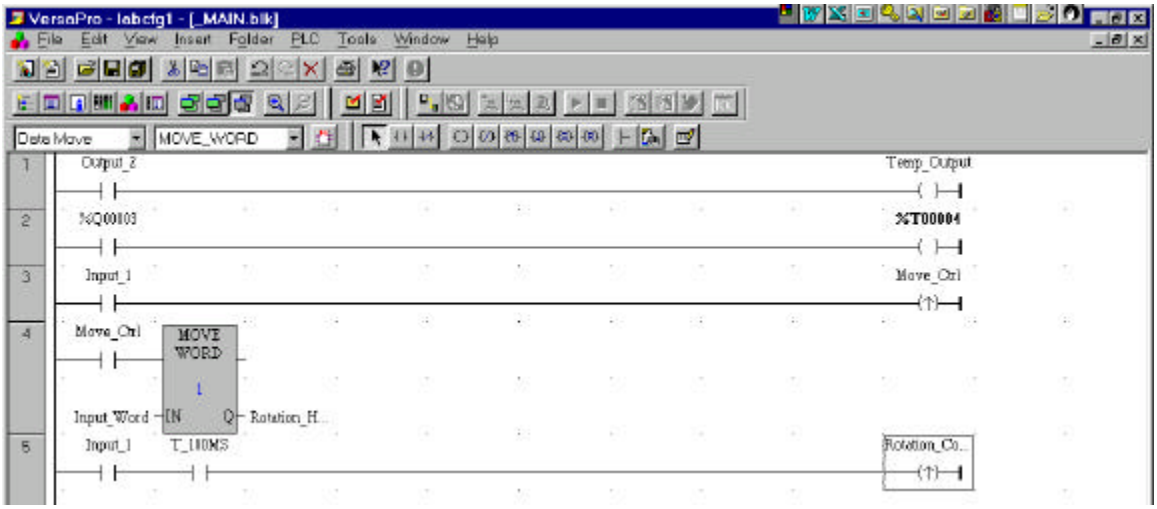
24. Click on the area to the left of the “IN” node and type in Input_Word and in the right area named “Q” type Rotation_Hold.



25. Create another rung below the previous one with a NORMALLY OPEN CONTACT labeled Input_1.

26. Enter a NORMALLY OPEN CONTACT to the right side of the rung . Double click on the contact and when the “pull-down” box appears, click on the arrow to the right of the box and choose T_100MS (System Variable) as its reference.

27. Enter a POSITIVE TRANSITION COIL to the right side of the rung with Rotation_ Control as its reference.

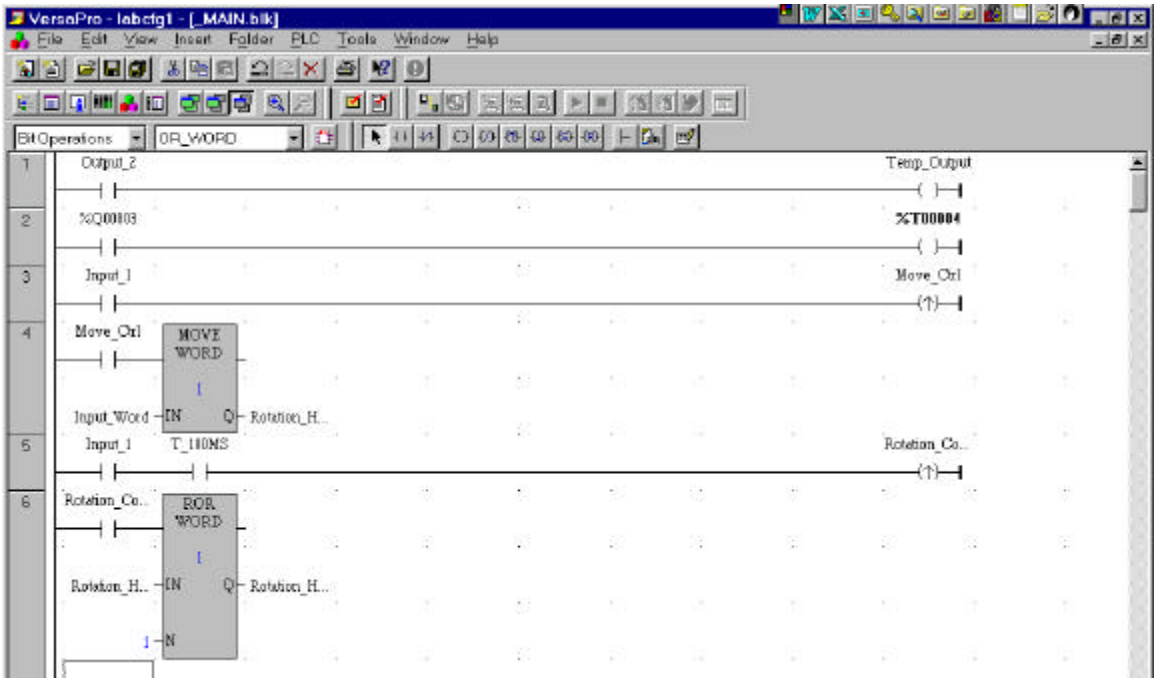


28. In a new rung enter a NORMALLY OPEN CONTACT. Double click on the contact and when the "pull-down" box appears, click on the arrow to the right of the box and select Rotation_ Control from the list and click below the box to enter the reference.

29. Enter a Bit Operations to the right of this contact by clicking on the arrow to the right of the Function Group toolbar and selecting BIT OPERATIONS. Do the same for the Function Name toolbar and select ROR_WORD.

30. Double click or enter on the ROR_WORD function to display the function properties. Type 1 in the length box.

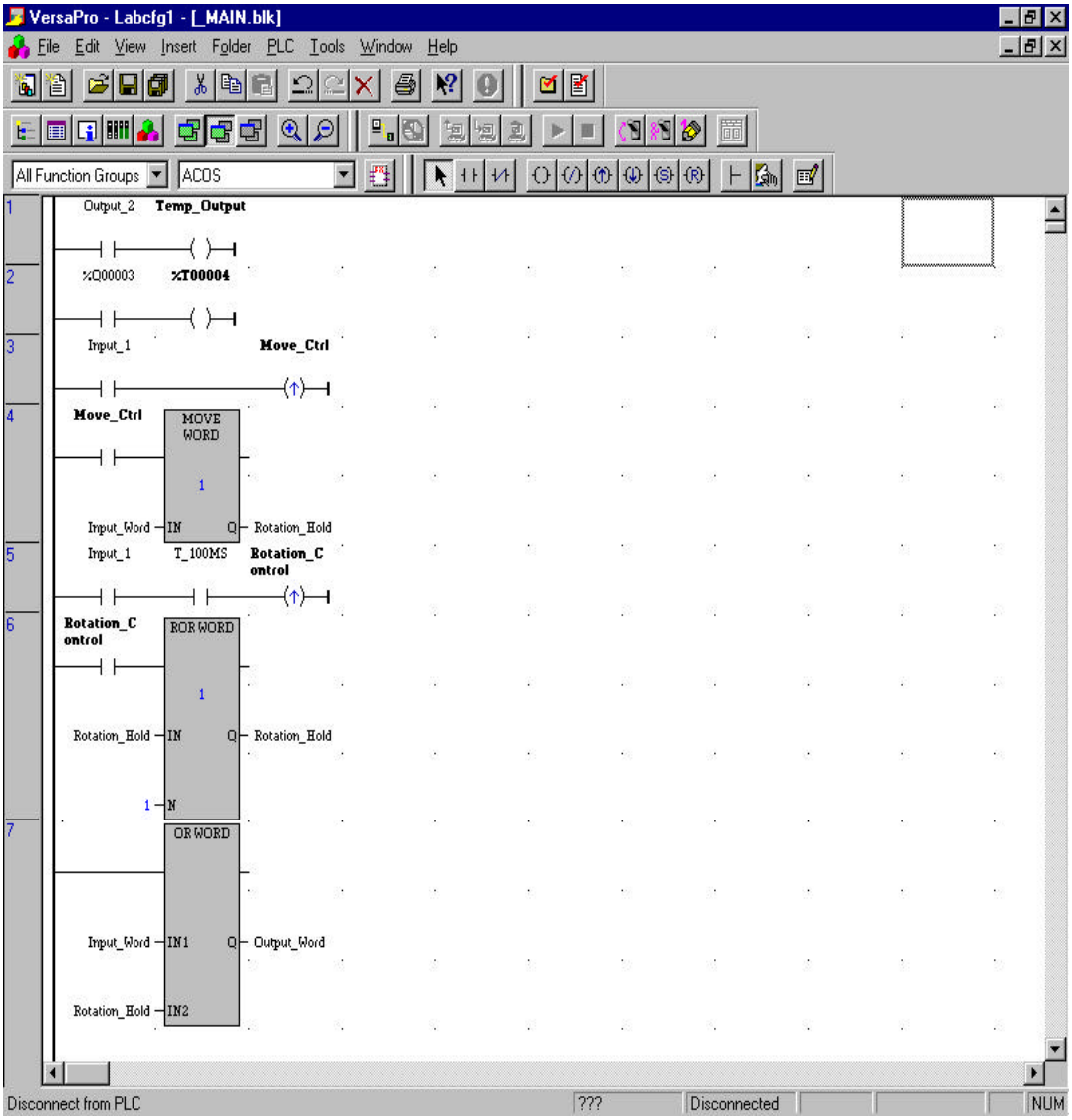
31. Click on the area to the left of the “IN” node and type Rotation_Hold and also for the right area named “Q”, for the “N” node type 1.



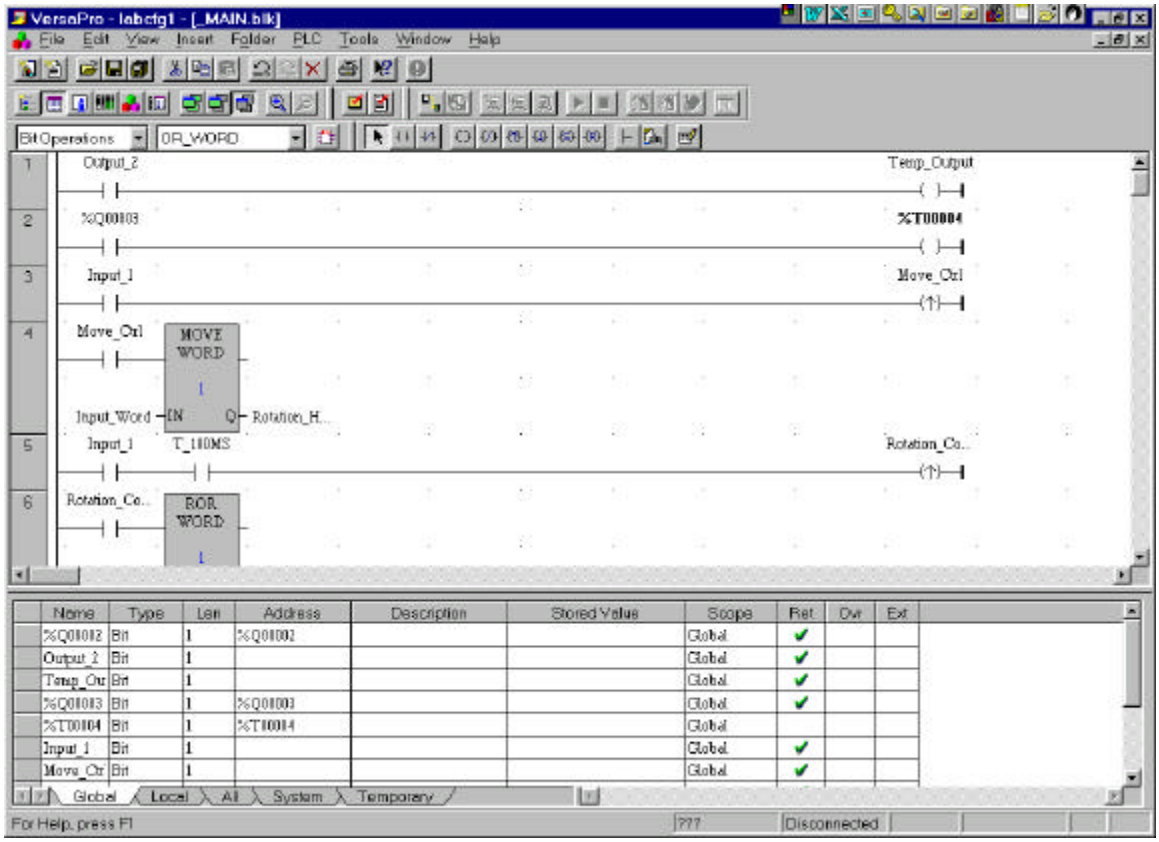
32. Insert another rung and enter a Bit Operations to the right of this contact by clicking on the arrow to the right of the Function Group toolbar and selecting BIT OPERATIONS. Do the same for the Function Name toolbar and select OR_WORD.

33. For the Input 1 “IN1” select Input_Word and for the Input 2 “IN2” select Rotation_Hold. Click on the “Q” area and select Output_Word.

The finished relay ladder logic should look similar to the workspace shown below.



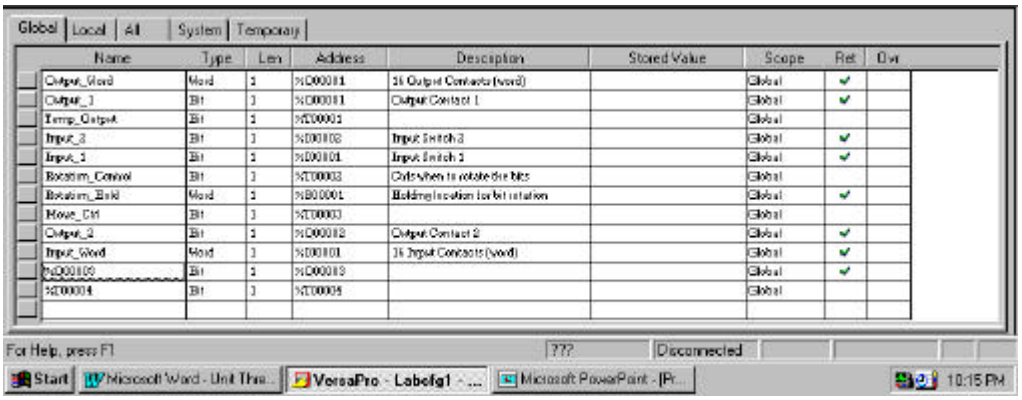
Locate the **VARIABLE TABLE** button on the toolbar and click it to show the Variable Declarations Table.



34. Double click on %Q00003 under the "Name" column and type Output_3. Press the **ENTER** key on the keyboard to enter "Output_3" as the NAME of the reference %Q00003.

35. Click under the "Description" column and type in the description Output Contact 3.

36. Repeat steps #34 and #35 for the other elements as follows:



When you are finished, the Variable Declaration Table should look similar to the picture above.

37. Click on the other tabs to see the other types of variables that are available on the system. Only the "Global", "All" and "System" tabs will have any data listed in them at this time.

38. When finished, close the Variable Declaration Table.

39. Click the **SAVE** button on the toolbar to store the newly created logic to the lab folder.

Once you've created logic, you can check either portions of it or its entirety for syntax errors. In a syntax check, instructions within your logic are compared against information about the PLC to which you intend to send the logic, and any unsupported instructions are brought to your attention. Such a check is automatically triggered when you store the logic to the PLC.

You can also manually initiate a syntax error check through either the Folder menu or the Folder Browser's Context-Sensitive menu, or the Check All or Check Selected Blocks toolbar buttons.

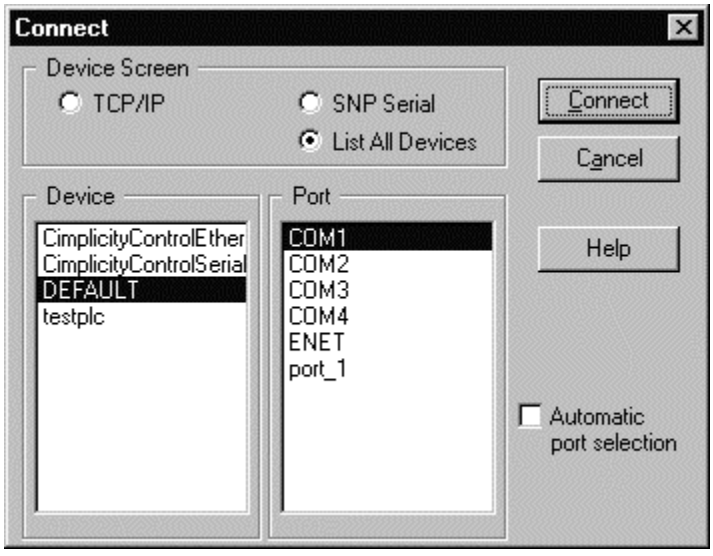
If the block or folder that you check contains invalid syntax, a list of errors needing correction displays as shown below

```
---------- Checking folder: Prueba. ----------

Checking block: _MAIN....
ERROR: [NOCON] Missing required input operand POWER_FLOW.
ERROR: Invalid dangling expression.
ERROR: [NOCON] Unexpected end of rung.
_MAIN - 3 error(s), 0 warning(s).

---------- Syntax check summary. ----------
Total errors: 3
```

Double clicking the error will take you to the rung with the problem

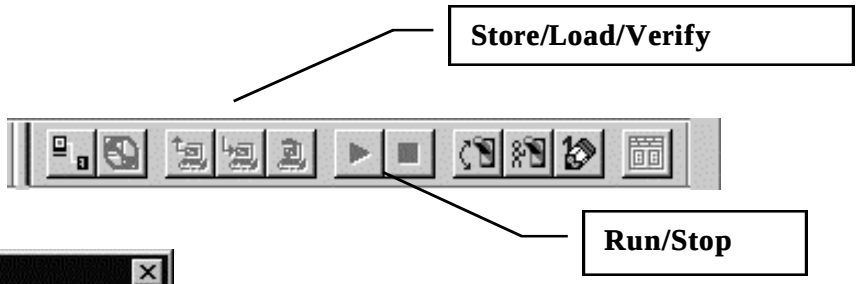
STORING THE CONFIGURATION TO THE VERSAMAX PLC

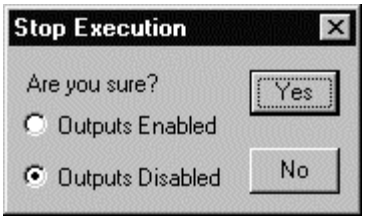
1. Select the **CONNECT** button on the tool bar or on the Menu bar, click on **PLC**, then **CONNECT**. A pop-up window will appear with **Device Screen**, **Device** and **Port** selections. Click on the **LIST ALL DEVICES** radio button and select the proper "Device" and "Port" for communications to the PLC.



2. Click on **CONNECT** and the VersaPro software will attempt to communicate with the PLC. Note the Status information at the bottom of the screen which will show the "connect" status and inform you when the operation is complete. If the connection is successful, a "Ready" along with the CPU Run Status, Scan time and whether or not the program is equal to the PLC.

3. Should an error occur, a pop-up will appear with an indication of the problem. Insure that the cabling is correct and that the PLC is using the same baud rate specified when the CPU was configured.

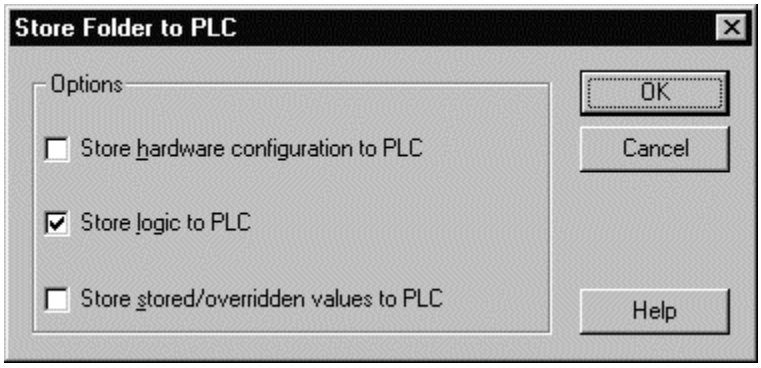




4. If the CPU is in the RUN mode, select the **STOP PLC** button on the toolbar or select **PLC**, then **STOP** on the menu bar. Click on either "Outputs Enabled or Disabled" and select **YES** to stop the PLC.

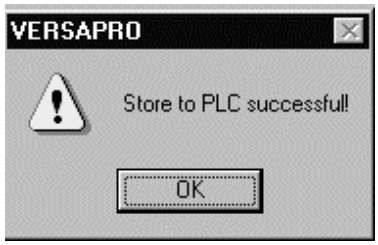
When the PLC folder name and your program Folder name are the same you can store the logic to the PLC without stopping the PLC.

5. Select the **STORE TO PLC** button on the toolbar or select **PLC**, then **STORE** on the menu bar.



6. Insure that only the **STORE LOGIC TO PLC** radio button is checked and select **OK** to begin the store operation.

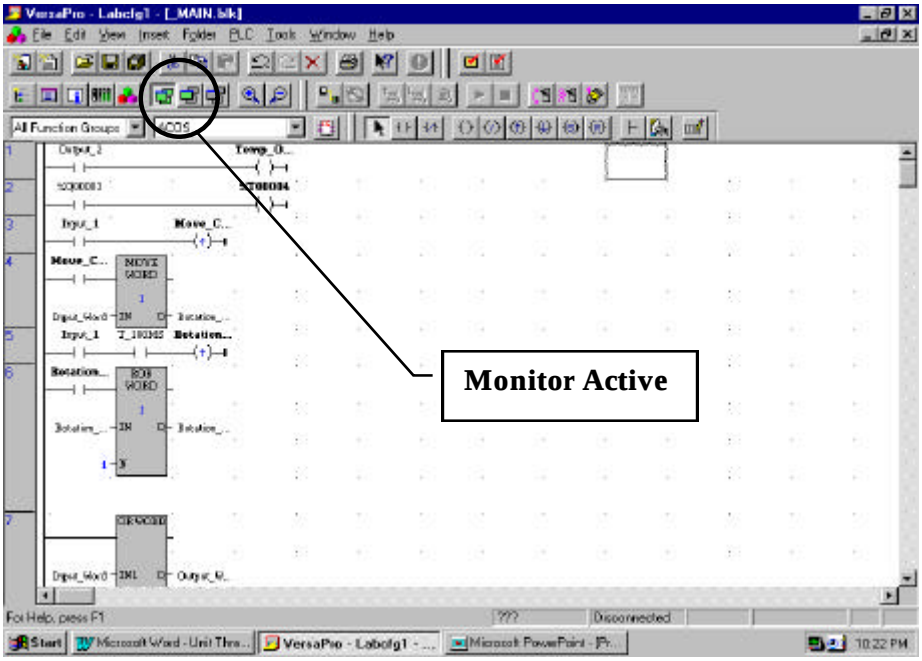
7. This message will appear when the store operation is complete.



8. Return the CPU to the **RUN** mode.

MONITORING THE LOGIC PROGRAM

1. Insure that the CPU is in the RUN mode.
2. Turn on the **MONITOR ACTIVE WINDOW** button on the toolbar to begin monitoring your program. Test your Inputs with the demo switches in order to monitor your program



Override and Toggle

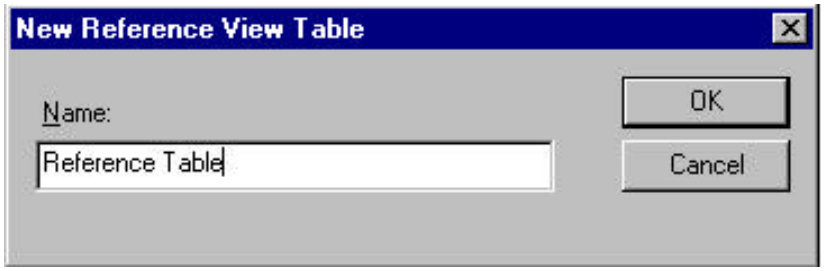
1. Move the cursor to the "Output_2" contact. Right-mouse click on the contact and click on OVERRIDE A line will appear under the contact name. Right-mouse click again and click on TOGGLE to change the state of the reference.
1. Click on **OVERRIDE** again to remove the Override. The contact should return to its actual state.
2. Try these operations on some of the other contacts and coils to see what effects they have on the program operation.

REFERENCE VIEW TABLES

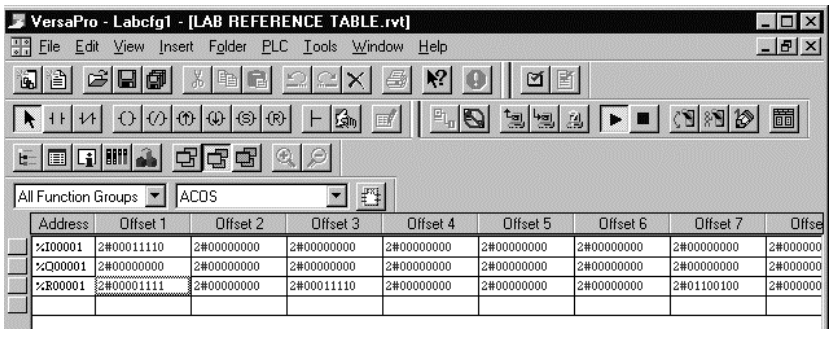
Reference View Tables (RVTs) contain a list of references that can be monitored and updated in real-time. A folder can have zero or more reference tables. RVTs can be cut and copied and dragged and dropped between Folders. More than one Reference Area can be entered into a single RVT.

The number of entries contained in an RVT does not affect the performance of the RVT. Performance is affected only by the number of entries that are displayed and have to be updated in the view.

1. Select **FILE** and then **NEW REFERENCE VIEW TABLE**. A window will appear on the workspace. Enter a name for the table and click **OK**.

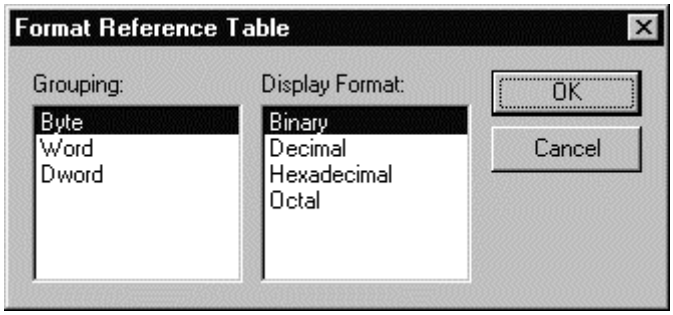


2. A blank Reference Table will appear on the screen. Click under the "Address" column and enter **%**I1 in the highlighted area. Press the **ENTER** key to enter the reference.
3. Enter %Q1 under the "Address" column in the next row and press **ENTER** again.
4. Do the same for next row, but enter %R1 as the reference you are going to monitor. When you are finished, the Reference Table should look similar to the picture below.

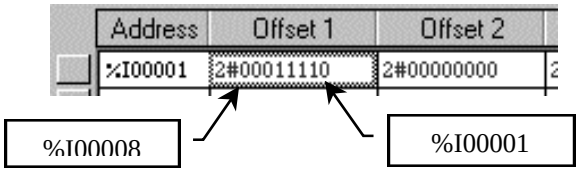


5. Note that all of the table values are displayed in Binary. You may change this for the ENTIRE table by right-mouse clicking anywhere on the table and selecting **FORMAT REFERENCE TABLE**.

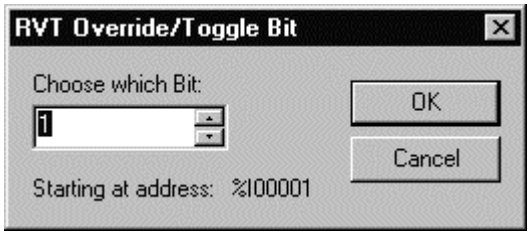
6. A window will appear as shown. Select the "Grouping" and "Display Format" desired by highlighting each selection, then click **OK**. The selected format applies to the entire table



7. In the "Binary" mode, the format of the bits is from right to left in each column as shown in the picture below.

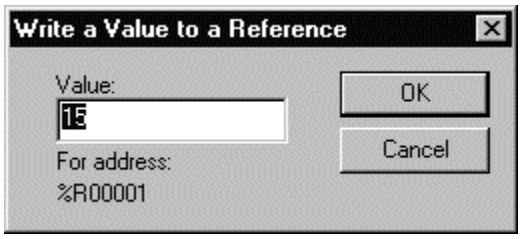


8. Overrides and Toggles may also be executed from this display. Right-mouse click on the desired group of bits and a window will appear allowing you to select a bit to **TOGGLE** or **OVERRIDE**. Click **OK** to execute the function.



9. If an **OVERRIDE** was done, a line will appear under the selected bit in the table. A **TOGGLE** operation will work in the same way it did in the Ladder Logic display.

10. In order to enter a value in a register, repeat Step #6 but select "Decimal" as the display format.



11. Right-mouse click on the desired register and when the **WRITE A VALUE TO A REFERENCE** window appears, type in the new value, then click **OK**.

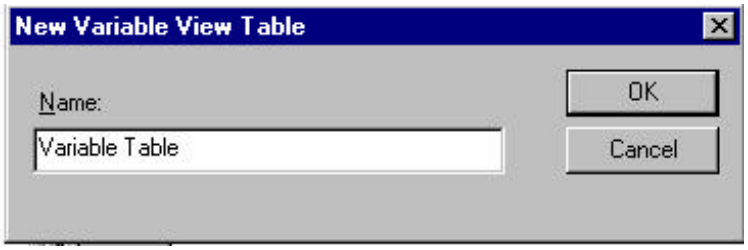
12. Experiment on some of the other input, output and register locations. Try switching some of the inputs on your simulator and viewing the results on the display.

13. Close the Reference View Table. You can create many Reference Tables and recall them at any time by selecting the **TOGGLE FOLDER BROWSER** button on the tool bar and double-clicking on the desired Reference Table name.

VARIABLE VIEW TABLES

Variable View Tables (VVTs) allow you to monitor the states of variables. A folder may have zero or more mixed-type VVTs. VVT entries can be cut and copied, and dragged and dropped between folders.

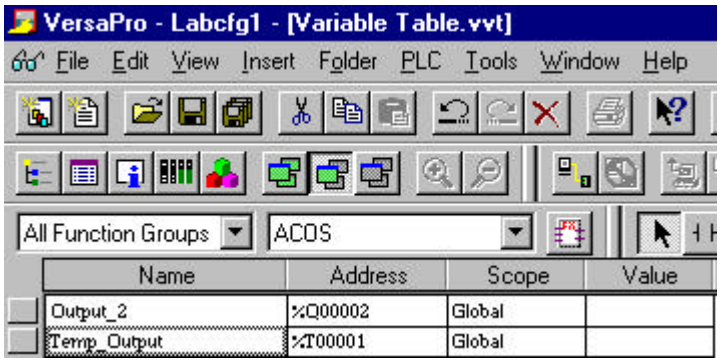
1. Select **FILE** and then **NEW VARIABLE VIEW TABLE**. A window will appear on the workspace. Enter a name for the table and click **OK**.



2. A blank Variable Table will appear on the screen. Double Click under the "Name" column and when the "pull down" box appears, click on the arrow to the right of the box and choose the variable "Output_2". Press the **ENTER** key to enter the variable.

3. The Variable Declaration Table information could be copied to the VVT selecting the variables you want to copy, right mouse button and select copy, paste the information in the Variable View Table.

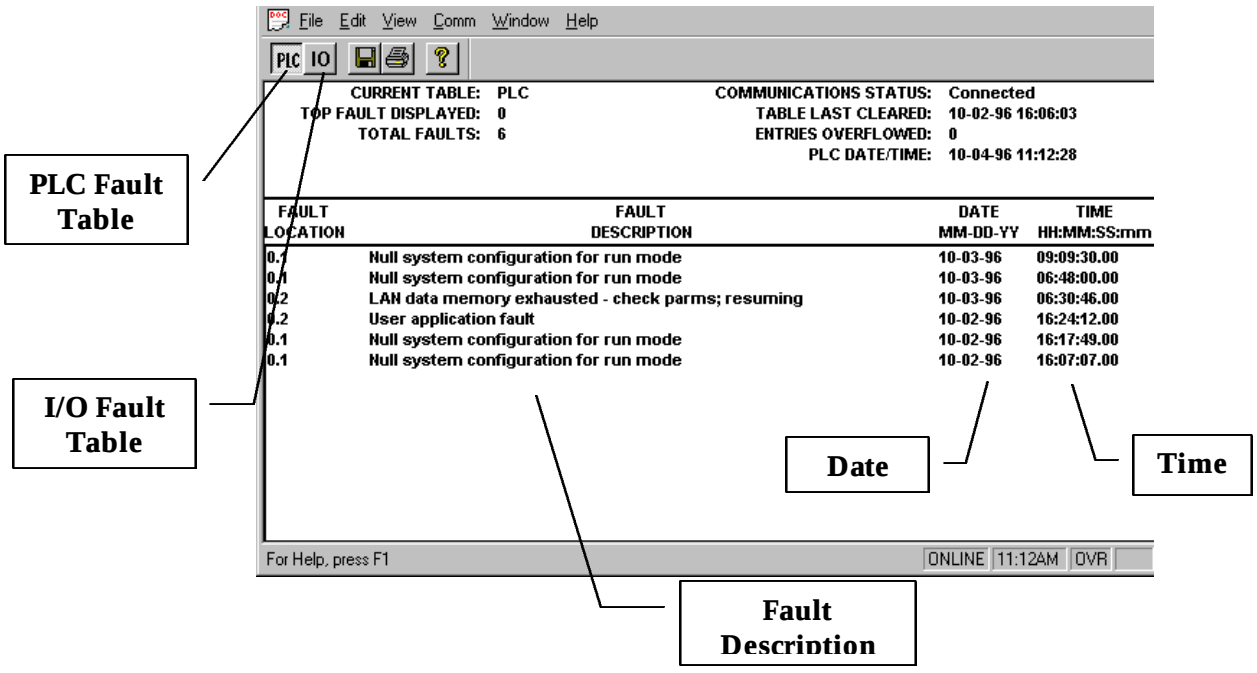
4. Do the same for next row, but enter "Temp_Output"as the state of the variable you are going to monitor. When you are finished, the Variable Table should look similar to the picture below.



FAULT TABLES

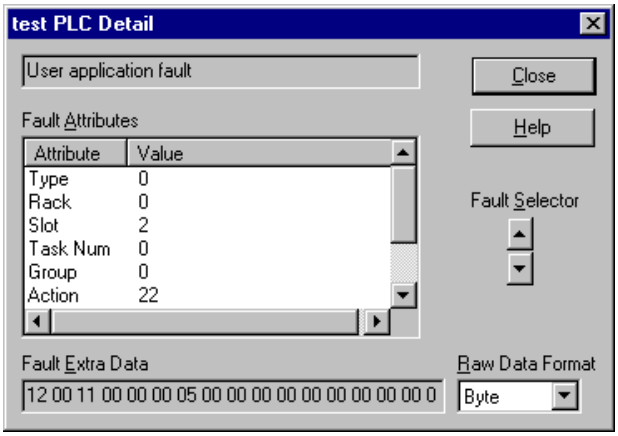
To view the PLC and I/O Fault Tables, your PC must be connected to and online with the PLC. You can connect to the PLC before entering the fault tables or go to the Comm menu with either table and select Connect to PLC.

1. To view the PLC Fault Table from VersaPro, go to the **TOOLS** menu, choose Faults, then choose CPU. If you are in the I/O Fault Table, click the PLC toolbar button to access the PLC Fault Table


File Edit View Comm Window Help
PLC IO
CURRENT TABLE: PLC
TOP FAULT DISPLAYED: 0
TOTAL FAULTS: 6
COMMUNICATIONS STATUS: Connected
TABLE LAST CLEARED: 10-02-96 16:06:03
ENTRIES OVERFLOWED: 0
PLC DATE/TIME: 10-04-96 11:12:28
FAULT LOCATION
FAULT DESCRIPTION
DATE MM-DD-YY
TIME HH:MM:SS:mm
0.1 Null system configuration for run mode 10-03-96 09:09:30.00
0.1 Null system configuration for run mode 10-03-96 06:48:00.00
0.2 LAN data memory exhausted - check parms; resuming 10-03-96 06:30:46.00
0.2 User application fault 10-02-96 16:24:12.00
0.1 Null system configuration for run mode 10-02-96 16:17:49.00
0.1 Null system configuration for run mode 10-02-96 16:07:07.00
For Help, press F1
ONLINE 11:12AM OVR
PLC Fault Table
I/O Fault Table
Date
Time
Fault Description

The fault table provides a summary description of each fault. Additional information describing the fault can be accessed:

1. To obtain additional information for a specific fault, double-click the fault. The Fault Detail dialog box will appear.



2. To list all faults and expanded fault data select the File menu and choose Save. Once you have saved this information, you can then use a text editor to view or print the text file created by the save operation.

You can clear the entire fault table by pressing F9 or selecting Clear Table from the Edit menu. Clearing faults also clears latched fault indications in all connected logged-in Genius blocks. Faults can also be cleared from program logic.

You can clear specific entries in the fault table by highlighting the entries and selecting Clear Entry from the Edit menu.
If you select Clear Table to clear the fault table, the fault bit in the PLC is reset, and you can put the PLC back into run mode. If you select Clear Entry, the fault bit is NOT reset, and you cannot place the PLC in run mode.

Clearing the fault table removes the faults. However, it does not fix the underlying problem that generated the fault. Clearing the fault table will allow the PLC to be put in Run mode, even though the fault condition may still exist. If the condition that caused the fault still exists, the fault may be reported again after storing the configuration, cycling power to the PLC, or during a Stop-to-Run transition.

CREATING, INSERTING BLOCKS

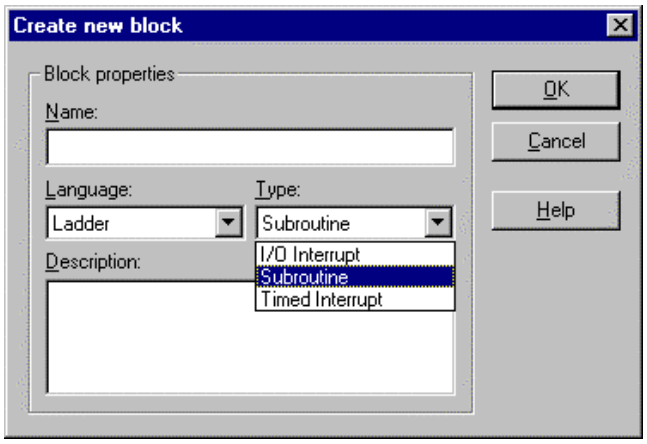
VersaPro allows you to use multiple blocks per folder, one of which must always be a _MAIN block. A block may be written in either Ladder Diagram (LD) or Instruction List (IL) programming language.

If you want to use multiple blocks in your program, you can create the blocks, which can be called from the _MAIN, or any other block. You can also place a call to a block that does not yet exist, but that you intend to create. When you compile, a syntax error displays reminding you that the new block has not been defined. More than one block can be open at once.

Creating a New Block

Creating a new block from the folder browser is quick and easy – just follow these steps:

1. With a folder open, select the File menu and choose New Block, or click the New Block button on the toolbar .The New Block dialog box will appear.



2. Enter a block name: “BLOCK”. A block cannot have the same name as another block that exists in the open folder.

3. Select Instruction List language for the new block using the drop-down list

4. Configure the Block Type

Note: Timed and I/O Interrupt Blocks are not supported in VersaPro Version 1.0.

5. Create a standard block selecting Subroutine

6. Enter a description for the block. You may enter up to 64 characters. This field is optional.

7. Click OK when you are finished filling in dialog box fields. VersaPro will create the block.

ENTERING INSTRUCTION LIST LOGIC

The Instruction List Editor is used to create programs using the Instruction List programming language. Instruction List is a text-based language, which consists of a series of instructions, which combined, represent actions to be performed on a PLC.

The IL Editor contains the following columns / fields:

- *Label* Displays a label that can be referenced by a JUMP instruction.

- *Instruction* Displays the function name. Instructions are entered by clicking on the Instruction field and then typing or selecting from the drop-down list.

- *Operand* Displays the variables or constants used by the instruction. Operands are entered by clicking on the field and then typing or selecting from the drop-down list. This list only displays variables appropriate to this type of instruction.

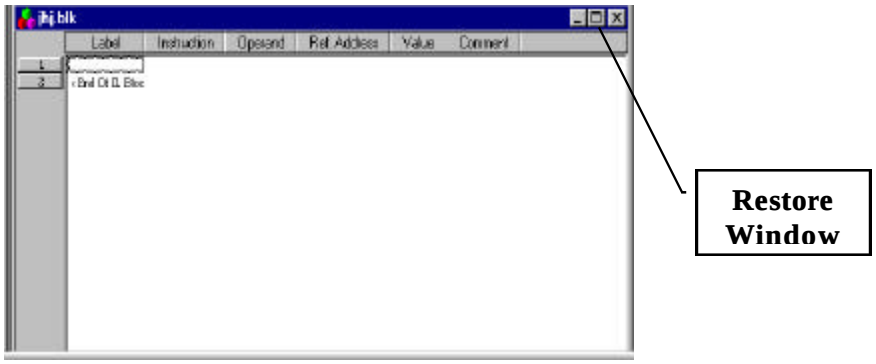
- *Reference Address* Displays the memory location of the operand.

- *Value* The actual value of the operand in the PLC.

- *Comment* Displays an optional comment about the instruction.

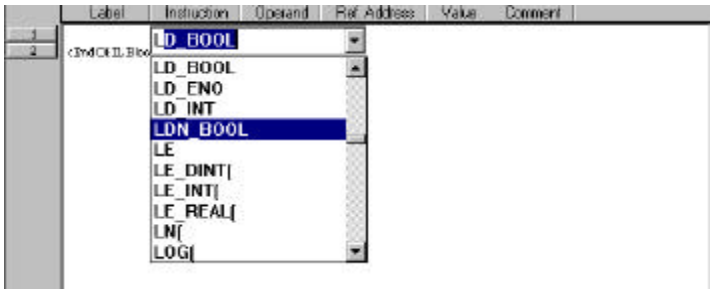
Instructions consist of program elements to perform boolean logic and execute functions such as timers, math functions and other functions. Variables and constants are assigned as operands for the instruction. Instructions are selected directly in the IL Editor window.

With the IL Editor workspace displayed, maximize it by clicking on the “Restore Window” button at the top right of the workspace.



You are now going to create a small Instruction List program, so follow the next steps carefully. All of the function buttons will display their functions when the cursor is placed on them.

1. Double-click on the Instruction cell and a pull down box will appear. Select the LDN_BOOL Instruction.



2. Double-click on the Operand cell and select ALW_ON, then press Enter.
3. In the next row select the instruction ST_BOOL.
4. Type OUTPUT in the “Operand” cell, and press Enter.
5. In the Variable Declaration Table define OUTPUT double clicking under the “Address” column and typing %Q00005.
6. Initiate a syntax error check, download the program to the PLC and monitor it.

Once a block is part of the Folder Browser, you can open and edit the block at any time. Multiple blocks may be open at the same time. To open a block in the Folder Browser, select the block you want to open and double-click the block name.

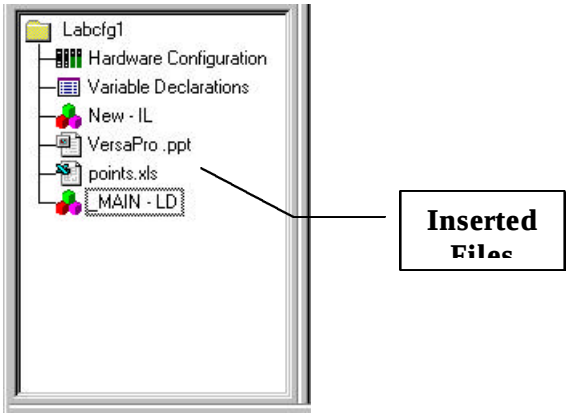
Functions are identified as FUNCTION_NAME. If a function is selected, parameters will appear between the brackets framing the function block.

INSERTING AND EDITING OBJECTS

VersaPro allows you to associate objects with your folder. For example, you can insert an Excel spreadsheet, Word document or Access database file into your folder. Once the object is associated with your folder, you can edit the object by double-clicking in the Folder Browser

To insert an object into a folder:

1. Open the folder into which you wish to **INSERT** an object. Go to the **FOLDER BROWSER**
2. Select the item below which you want to insert an object.
3. Select the Insert menu and choose Object, or click the right mouse button and select **INSERT OBJECT**. The Insert Object dialog box will appear.
4. Select the name and the location of the file you want to insert. VersaPro component files (files with an extension of .blk, .fld, .hwc, .wvr, .wrt, or .vdt) may not be inserted
5. Click **OPEN**. The object is inserted.



To launch an inserted object:

1. Open the folder containing the object that you want to launch. Go to the **FOLDER BROWSER**.
2. To open the object, double-click on the inserted object, or select the object using the up or down arrow keys and then press **ENTER**, or select the object, click the right mouse button and choose **OPEN** .

If an application is associated with the folder type, the application will start up, opening the selected object. If no application is associated with the object, the standard "Open With" Windows prompt will appear.

VersaPro provides a flexible browser interface to manage the blocks, view tables, variable declaration tables and objects associated with the folder.

PRINTING THE PROGRAM LOGIC

VersaPro provides an interface, which allows you to customize your printout and print your program. You can customize your printout through the Print Report dialog box.

- *Printing Blocks.* You can select to print all or selected blocks through the Print Blocks checkbox, you can also include logic, block properties and Variable Information.

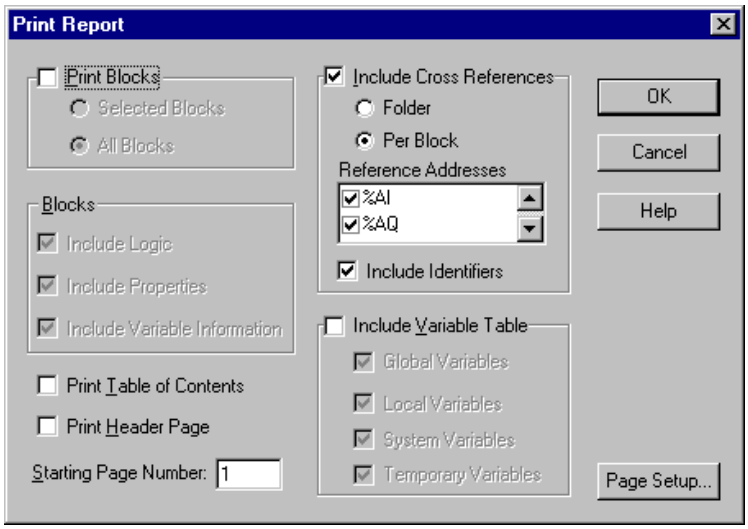
- *Printing TOC and Header Page.* To include a Table of Contents and/or a header page, check the appropriate boxes. You may use the default starting page number (1), or change the number, as required for your documentation.

- *Including Cross Reference:* Check the Include Cross References box to incorporate cross reference information into your listing. You can build cross references across the entire folder, or on a per block basis.

- *Including the Variable Table* To include variable tables in your listing, click the Include Variable Table checkbox. Select the types of variables you want to include in the variable listing: Global Variables, Local Variables, System Variables and Temporary Variables

Perform the following steps to customize and print your program:

1. From the open folder, select the File menu and choose Print Report. The Print Report dialog box will appear. Use this dialog box to customize your listing as described below.



2. Click the **OK** button to print your program.

PRINTING THE CONFIGURATION

You can print your Rack configuration by going to the File menu and choosing Print. To set print options, click the Range button in the Print dialog box. The following print options are available:

1. You can choose to print hardware configuration information for the entire rack, or for a range of modules (selected by rack:slot).
2. You can select to print overview or detail for the configuration:

- *Overview* Prints the Rack as shown on the screen.
- *Detail* Prints the parameter information for each module in the Rack.
- *Reference Details* Prints a table showing start and end memory locations, rack slot addresses, and other reference data.

If you want to change your target printer or paper size or orientation, click the Setup button in the Print dialog box.

Appendix A

This appendix shows the menu bars and toolbars used in VersaPro Programming Software

VERSAPRO MENU BARS

File Menu

The file menu includes standard file and print operations for VersaPro components.

Option	Button	Action
New Block (Ctrl + B)		Creates a new block.
New Reference View Table		Creates a New Reference View Table
New Variable View Table		Creates a new Variable View Table
Save (Ctrl + S)		Saves the selected item.
Save All		(Shift + Ctrl +A) Saves the entire folder.
Close (Ctrl + F4)		Closes the selected window.
New Folder (Ctrl + N)		Creates a new folder, or imports an existing folder.
Open Folder (Ctrl + O)		Opens an existing folder.
Close Folder		Closes the open folder.
Print		Prints the selected item.
Print Report		Allows you to define which components and blocks should be included in a report, then print the report.
Page Setup		Allows you to specify page parameters.
<MRU list>		Lets you open a recently used folder from a list of Most Recently Used folders (the MRU list) that appears here on the menu.
Exit (Alt + F4)		Closes VersaPro.

Edit Menu

Option	Button	Action
Undo <action>		(Ctrl + Z) Reverses the previous action. The previous action (for example, Drop or Delete) appears instead of "<action>".
Redo <action>		(Ctrl + Y) Reverses the previous undo action. The previous action (for example, Drop or Delete) appears instead of "<action>".
Cut (Ctrl + X)		Removes the selected item and places it on the Clipboard.
Copy (Ctrl + C)		Copies the selected item to the Clipboard.
Copy As		Copies the selected element(s) as either a bitmap image or as text to the Clipboard.
Paste (Ctrl + V)		Pastes the Clipboard contents to the selected area.
Delete (Del)		Deletes selected item(s).
Delete Row (Ctrl + D)		Deletes selected row(s).
Select All (Ctrl + A)		Selects all items
Find In Blocks		Allows you to find a particular variable Name or Address, or a Call to a Subroutine in blocks.
Find/Replace		Allows you to Find and Replace a variable Name or Address, a Call to a Subroutine, or a Jump/Label or MCR/END_MCR pair in logic
Find Next (F3)		Finds next item of defined search criteria.
Go to (Ctrl + G)		Allows you to go to a row in IL logic or a rung in LD logic.
Go to Variable (Ctrl + F11)		Allows you to got to a variable in the VDT.
Properties (Alt + Enter)		Allows you to add a description to the current block, to add a description to the current folder, or to set the properties of temporary variables or conversion variables.

View Menu

Option	Button	Action
Toolbars (Ctrl + T)		Shows or hides toolbars.
Function Toolbars		Allows you to expand or compact function toolbars
Status Bar (Alt + O)		Shows or hides the Status bar.
Folder Browser		(Alt + 1) Opens or closes the Folder Browser window.
Information Window(Alt + 2)		Opens or closes the Information window.
Variable Declaration Table (Alt + 3)		Opens or closes the Variable Declaration Table Window
Hardware Configuration (Alt + F4)		Opens the Hardware Configuration window.
MAIN block (Alt + F5)		Opens the _MAIN block.
Sort		Allows you to sort specific columns of the VDT in ascending or descending order.
Monitor		Displays real-time logic execution on the PLC.
Display Format		Allows you to change the display format of monitored logic.
Format Reference Table		Allows you to determine how much data is displayed in each Reference View Table row.
Zoom In (Ctrl + "+")		Magnifies the selected item.
Zoom Out (Ctrl + "-")		Decreases the magnification of the selected item.

Insert Menu

Option	Button	Action
Object		Allows you to insert an object into the Folder Browser.
Block		Allows you to insert a block into the Folder Browser.
Row (Ins)		Inserts a row above the selected row in the IL editor, LD editor, VDT, or a View Table.
Comment		Allows you to insert a comment.

Folder Menu

Option	Button	Action
Check Selected Block(s)		(Ctrl + F7) Checks the selected block(s) for syntactic correctness. If no block is selected, this menu option reads "Check Block_Main" and selecting it results in checking the _Main block for syntactic correctness.
Check All (F7)		Checks all blocks for syntactic correctness.
Lock/Unlock		Locks or unlocks the open folder.
Backup (Ctrl + F8)		Backs up a folder.
Restore (F8)		Restores a folder.

PLC Menu

Option	Button	Action
Connect (F9)		Connects to the PLC
Disconnect (Ctrl + F9)		Disconnects from the PLC
Store (Alt + F2)		Stores the contents of a folder to the PLC
Load (Ctrl + F2)		Loads the contents of a folder from the PLC
Verify (F4)		Verifies the equality of the current folder and the connected PLC.
Clear		Clears the memory of the PLC.
Flash/EEPROM		Allows you to read, write, or verify Flash memory.
Run (F5)		Runs the logic resident in the PLC
Stop (Ctrl + F5)		Stops the logic resident in the PLC
Toggle (F12)		Allows you to toggle a reference.
Override (F11)		Allows you to override a reference.
Write Reference Value		Allows you to write a value to a reference.
Tuning Parameters		Allows you to tune PID Instructions.
Status Info (Shift + F4)		Allows you to view the status of the PLC.
Abort! (F10)		Stops a communications action.

Tools Menu

Option	Action
Fault table	Opens a fault table.
Communications Setup	Allows you to define communications parameters
Import Variables	Allows you to import a variable.
Export Variables	Allows you to export a variable.
Convert Block	Allows you to convert logic from one language to another.
Options (Ctrl + E)	Allows you to set General (language–related) and Display (colors, fonts), Ladder (show fields, cell width), and Autoconnect options.

Window Menu

Option	Action
Cascade (Alt + 6)	Arranges all open block windows so that all title bars are visible and the active window is in front.
Tile Horizontally (Alt + F7)	Resizes and horizontally arranges all open block windows so that all of them are visible.
Tile Vertically (Alt + F8)	Resizes and vertically arranges all open block windows so that all of them are visible.
Arrange Icons (Alt + F9)	Aligns all minimized block windows.
Close All (Alt + F10)	Closes all open block windows.
Next Window (F6)	Makes the next window in the clockwise direction active.
Previous Window (Shift + F6)	Makes the next window in the counter-clockwise direction active.
[Current Window]	Displays the name of the active window.

Help Menu

Option	Action
Contents and Index	Displays VersaPro Help.
About VersaPro™	Displays information about this version of VersaPro and about available memory and disk space.

TOOLBARS

VersaPro has six toolbars: the Standard toolbar, the Ladder toolbar, the View toolbar, the Folder toolbar, the PLC toolbar, and the Function toolbar. You may hide any or all toolbars to provide more space to work. Unavailable toolbar buttons appear in dim gray.

Standard toolbar

The Standard toolbar is displayed by default, although you can hide it. The Standard toolbar buttons provide common functions, such as opening folders, saving folders and blocks, cutting, copying and pasting, and undo and redo.

Button	Function
	Creates a new folder.
	Creates a new block.
	Opens an existing folder.
	Saves the selected block.
	Saves the entire folder.
	Removes the selected item and places it on the Clipboard.
	Copies the selected item to the Clipboard.
	Pastes the Clipboard contents to the selected area.
	Reverses the previous action.
	Reverses the previous undo action.
	Deletes selected item(s).
	Sends the selected item to a printer.
	Displays context-sensitive help.
	Abort.

Ladder toolbar

Use the Ladder toolbar to create ladder logic using "click and drop."

Button	Function
	Selects the normal pointer; disables "click and drop"
	Lets you drop a normally open contact
	Lets you drop a normally closed contact
	Lets you drop a normally open coil
	Lets you drop a normally closed coil
	Lets you drop a positive transition coil
	Lets you drop a negative transition coil
	Lets you drop a set coil
	Lets you drop a reset coil
	Lets you drop a vertical or horizontal wire
	Lets you drop a CALL instruction
	Lets you insert a comment row.

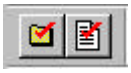
Note: Pressing the Escape key disables drop mode and makes the pointer active again.

View toolbar

Use the View toolbar to show or hide windows, to open the _MAIN block, or to magnify or reduce a window's contents. Buttons not related to the current task are not available.

Button	Function
	Shows or hides the Folder Browser
	Shows or hides the Variable Declaration Table
	Shows or hides the Information Window
	Opens the Hardware Configuration window
	Opens the _MAIN block
	Monitors all windows.
	Monitors the active window.
	Turns monitoring off.
	Zooms In on (magnifies) the active window's contents
	Zooms Out from (reduces) the active window's contents

Folder toolbar



Use the Folder toolbar to check syntax.

Button	Function
	Checks all blocks for syntactic correctness.
	Checks selected block(s) for syntactic correctness.

PLC toolbar

Use the PLC toolbar to connect to a PLC, and to perform operations involving a connected PLC, like storing and loading, or running and stopping logic.

Button	Function
	Connects to the PLC.
	Disconnects from the PLC.
	Stores this folder to the PLC.
	Loads a folder from the PLC.
	Verifies equality between the current folder and elements present on the PLC.
	Runs the PLC.
	Stops the PLC.
	Toggles a reference.
	Overrides a reference.
	Writes a value to a reference.
	Shows the status of the PLC.

Function toolbar

Use the Function toolbars to select function block types while programming in the RLD editor. The Function toolbars can be configured to display in expanded or compact mode. To change the mode, select the View menu, choose Function Toolbars, then select Expanded or Compact.